Вокруг теоремы Кэли

Задачу представляют Бурсиан О., Кохась Д., Кохась К.

Версия 0.99

Этот проект посвящен теореме Кэли о числе деревьев. Дерево — это связный граф без циклов. Теорема утверждает, что число помеченных деревьев с n вершинами равно n^{n-2}. Известно много доказательств этой теоремы, нашей целью будет познакомиться с некоторыми из них и изучить приложения различных подходов к этой теореме.

В каждом проекте есть довольно сложные задачи. По правилам конференции задачи можно решать, объединившись в команды; задачи разных проектов можно решать в составе разных команд. Мы советуем вам найти попутчиков, с которыми вам было бы интересно проводить исследования, описанные в этом проекте.

Задачи для первого знакомства

В олимпиадных задачах обычно используются графы с «обезличенными» вершинами — например, вершины — это какие-то города, а ребра — дороги между ними, или вершины — это люди (обычно безымянные), а ребра — это знакомства и т.п. В задачах о перечислении графов принята другая точка зрения — все вершины графа должны быть «индивидуальны». Чтобы не было диссонанса в терминологии, мы введем понятие «помеченный граф».

Пусть $[n] = \{1, 2, \ldots, n\}$. Дерево (или вообще произвольный граф) на n вершинах, вершины которого пронумерованы числами от 1 до n, называется *помеченным деревом* (соответственно *помеченным графом*). Чтобы построить помеченное дерево, можно взять дерево и расставить в его вершинах числа, или наоборот: мы можем считать, что множество $[n]$ — это множество вершин, и мы рисуем граф-дерево, соединяя эти вершины ребрами. Множество помеченных деревьев на n вершинах обозначим через $\mathcal{T}_n$.

Два (непомеченных) графа G_1 и G_2 с множеством вершин V_1 и V_2 называются *изоморфными* (или, попросту говоря, считаются одинаковыми), если существует такое взаимно однозначное отображение $f\colon V_1 \to V_2$, что вершины A, $B \in V_1$ соединены ребром в G_1 тогда и только тогда, когда $f(A)$ и $f(B)$ соединены ребром в G_2. Например, любое дерево с четырьмя вершинами изоморфно либо дереву «куриная лапа», либо дереву «путь длины три». В случае, когда G_1 и G_2 — помеченные графы, $V_1 = V_2 = [n]$ и в качестве f в этом определении берут тождественное отображение.

Мы предполагаем, что при решении задач разделе «Задачи для первого знакомства» теорема Кэли не должна использоваться. В формулировках задач через T_n обозначается количество помеченных деревьев на n вершинах, и вопрос о нахождении этой величины не ставится. Нумерация задач дана в соответствии с последующими разделами.

1.1. Граф называется *уницикличeским*, если он связный и содержит ровно один цикл. Докажите, что помеченных деревьев со 100 вершинами больше, чем помеченных уницикличeских графов с 98 вершинами.

1.2. Постройте взаимно однозначное соответствие между множеством, которое состоит из всевозможных отображений множества $[n]$ в себя, и множеством помеченных деревьев на n вершинах, в которых одна вершина снабжена красной меткой и одна — синей меткой (может оказаться, что обе метки поставлены возле одной и той же вершины).

1.3. Существует n^{n-1} различных отображений из множества $[n-1]$ в $[n]$. Докажите тождество

$$\sum_{j=1}^{n} C_{n-1}^{j-1}(n-j)^{n-j}T_j = n^{n-1},$$

разбив множество этих отображений на n частей так, чтобы j-е слагаемое оказалось равно количеству отображений в j-й части.

3.1. Дерево с n вершинами, ребра которого пронумерованы числами от 1 до $n-1$, называется *реберно помеченным.* Например, существует 4 различных реберно помеченных дерева на 4 вершинах — см. рис. 1. Докажите, что при $n \geqslant 3$ количество различных реберно помеченных деревьев на n вершинах равно $\frac{1}{n}T_n$.

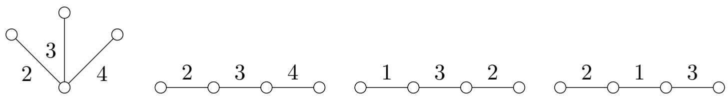


Рис. 1. Реберно помеченные курлапа и пути длины 3

Если мы выделили в дереве (помеченном или нет) одну из вершин, то будем называть такое дерево *корневым*, а саму отмеченную вершину будем называть *корнем.* Если же нам потребуется подчеркнуть, что ни одна из вершин нами не выделена, будем называть такое дерево *свободным. Листом* дерева называется вершина степени 1, за исключением случая, когда дерево корневое и корень имеет степень 1, в этом случае корень листом не считается. *Лес* — это граф, в котором каждая компонента связности является деревом.

Обозначим через $\mathcal{F}_n^k$ множество лесов на множестве вершин $[n]$, состоящих из k корневых деревьев с корнями 1, 2, …, k, причем таких, у которых вершина n находится в дереве с корнем 1 (рис. 2).

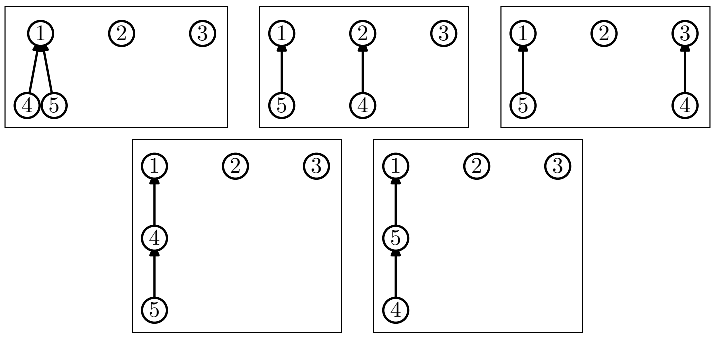


Рис. 2. Множество лесов $\mathcal{F}_5^3$. Мы ориентировали ребра «от периферии к корню»

Множества мы обозначаем «рукописными» буквами. Число элементов множества обозначается той же буквой в курсивном начертании. Так число в множестве $\mathcal{F}_n^k$ будем обозначать F_n^k.

1.4. Докажите, что при $2 \leqslant k \leqslant n-1$ выполняется рекуррентное соотношение

$$F_n^{k-1} = nF_n^k.$$

1.5. Пусть $V_1 = [r]$, $V_2 = \{r+1, \ldots, r+s\}$, $V = V_1 \cup V_2 = [r+s]$. Обозначим через $\mathcal{F}_{r,s}^k$ множество лесов, состоящих из k корневых деревьев с множеством вершин V с корнями 1, 2, …, k, у которых вершина $r+1$ находится в дереве с корнем 1, а у каждого ребра одна вершина из множества V_1, а другая — из множества V_2. Придумайте рекурсию (по k) для величины $F_{r,s}^k$

Вдоль улицы с односторонним движением расположено n мест для парковки автомобиля. На улицу въезжают последовательно n автомобилей с номерами от 1 до n в порядке возрастания. Каждый водитель едет к своему любимому месту парковки и, если оно не занято, припарковывается; в противном случае он едет дальше до первого свободного места и припарковывается там, если же все места дальше заняты, он уезжает (насовсем). Последовательностью предпочтений назовем список a_1, a_2, …, a_n любимых мест парковки первого, второго, …, n-го водителя.

4.1. Докажите, что существует $(n+1)^{n-1}$ последовательностей предпочтений, для которых все водители сумеют припарковаться.

4.2. Докажите, что успех или неуспех парковки на самом деле не зависит от порядка, в котором прибывают машины.

Следующая задача лежит в стороне от нашего сюжета. Но она позволяет понять, какие затруднения могут возникнуть, если мы станем перечислять непомеченные деревья.

3.2. Докажите, что количество различных (т. е. неизоморфных друг другу) непомеченных деревьев с n вершинами меньше 4^n.

1 Рекурсии, тождества, биекции

1.6. Докажите при помощи комбинаторных рассуждений (интерпретируя числа T_i как количества деревьев), что при $n > 1$ a) $T_n = \dfrac{n}{2}\sum\limits_{k=0}^{n-2} C_{n-2}^k T_{k+1} T_{n-k-1}$;

b) Сведите эту формулу, а также формулы из задач 1.3 и 5.1 a) друг к другу алгебраически.

1.7. Обозначим через $\mathcal{T}(n,k)$ множество помеченных корневых деревьев с n вершинами, в которых корень — вершина 1 — имеет степень k. Докажите, что

$$(n-1)(k-1)T(n,k) = (n-k)T(n,k-1).$$

«Треугольным деревом» назовем граф, определеямый по индукции следующим образом. Самое маленькое треугольное дерево вопреки своему названию — это полный граф на двух вершинах. Если уже задано какое-нибудь треугольное дерево, мы можем взять любое ребро AB в нем, взять новую вершину C и добавить к дереву вершину C и ребра AC, BC. Помеченным треугольным деревом будем называть треугольное дерево, у которого вершины пронумерованы числами от 1 до n.

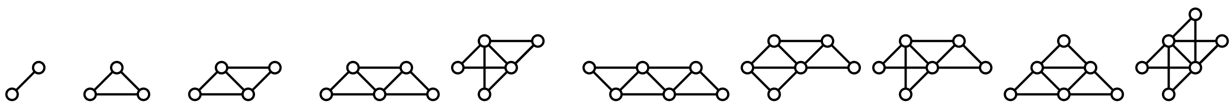


Рис. 3. Непомеченные треугольные деревья на 2, 3, 4, 5 и 6 вершинах.

Например, для $n = 5$ имеется 2 непомеченных треугольных дерева (рис. 3) и 70 помеченных.
Если некоторое ребро треугольного дерева выделено, то такое дерево будем называть «корневым».

1.8. Обозначим через $\Delta(n,k)$ число корневых помеченных треугольных деревьев на n вершинах, у которых корневое ребро принадлежит k треугольникам. Придумайте рекурсию (по k) для величины $\Delta(n,k)$.

2 Код Прюфера

Код Прюфера сопоставляет дереву с занумерованными вершинами последовательность его вершин следующим образом. Код Прюфера дерева с двумя вершинами — пустое слово. Если количество вершин дерева T больше двух, то обозначим через v лист с минимальным номером, а через u вершину, смежную с v. Тогда код Прюфера дерева T получается из кода Прюфера дерева $T - v$ приписыванием слева вершины u.

Проверьте, что Вы умеете решать задачу 2.1, и зарегистрируйте плюсик у жюри.

2.1. a) Найдите код Прюфера дерева с вершинами 1, 2, …, 10 и рёбрами (8,9), (8,4), (4,10), (10,3), (3,5), (10,6), (10,1), (1,7), (1,2).

b) Восстановите дерево по коду Прюфера 1, 1, 2, 5, 4, 2, 7.

c) Докажите, что код Прюфера определяет взаимно однозначное соответствие между множеством деревьев с данными n вершинами и множеством слов длины $n-2$ из этих вершин.

d) Докажите, что в коде Прюфера вершина степени d встречается $d-1$ раз.

2.2. Чему равно количество корневых помеченных деревьев с n вершинами, у которых вершина n является листом (и тем самым она не корень)?

2.3. Чему равно количество свободных помеченных деревьев с $n > 10$ вершинами, у которых степень вершины 1 равна 10?

2.4. Найдите количество помеченных унициклических графов с n вершинами, имеющих цикл длины k.

2.5. Обозначим через $S(n,k)$ число Стирлинга второго рода, по определению оно равно количеству разбиений множества $[n]$ на k непустых частей. Докажите, что количество помеченных деревьев с n вершинами, ровно r из которых — листья, равно $\frac{n!}{r!}S(n-2,n-r)$.

2.6. Обозначим через $\tau(k_1,k_2,\ldots,k_n)$ число помеченных свободных деревьев с n вершинами, у которых степень i-й вершины равна k_i+1. Докажите, что $\tau(k_1,k_2,\ldots,k_n)=\dfrac{(n-2)!}{k_1!k_2!\ldots k_n!}$.

3 *Результаты*

В следующих задачах можно сдавать несколько решений, если они принципиально отличаются.

3.3. Выведите теорему Кэли из задач: a) 1.4, b) 1.6, c) 1.7, d) 2.6.

3.4. Докажите, что количество лесов, с вершинами в множестве $[n]$, состоящих из k корневых деревьев, равно $C_{n-1}^{k-1}n^{n-k}$.

3.5. Докажите, что количество лесов, с вершинами в множестве $[n]$, состоящих из двух некорневых деревьев, равно $\frac{1}{2}n^{n-4}(n-1)(n-6)$.

3.6. Докажите, что количество остовных деревьев в полном двудольном помеченном графе $K_{r,s}$ с долями $V_1=[r]$ и $V_2=\{r+1,\ldots,r+s\}$ равно $r^{s-1}s^{r-1}$.

3.7. Пусть Δ_n — число помеченных треугольных деревьев с n вершинами, Λ_n — число корневых помеченных треугольных деревьев с n вершинами и корневым ребром 1–2.

a) Докажите, что $\Lambda_n=(2n-3)^{n-3}$. b) Найдите Δ_n.

Пусть в ряд расположено n различных предметов. *Циклическая* перестановка — это перестановка, перемещающая предметы по некоторому циклу: один предмет ставится на место другого, этот другой — на место третьего, и т. д., последний ставится на место первого. *Транспозиция* (ij) — это перестановка, меняющая местами предметы на i-м и j-м месте. Если считать, что множество $[n]$ является множеством вершин некоторого графа, то транспозицию (ij) можно интерпретировать как ребро, соединяющее вершину i с вершиной j.

Результат последовательного выполнения перестановок s_1, s_2, …, s_{n-1} определяет перестановку, которая называется *произведением* этих перестановок и обозначается $s_1s_2\ldots s_{n-1}$. Произведения, отличающиеся порядком сомножителей, мы считаем различными. Например, если перестановка s — это транспозиция, меняющая местами предметы на первом и втором месте, а перестановка t — это транспозиция, меняющая местами предметы на третьем и четвертом месте, то произведения st и ts определяют одну и ту же перестановку предметов, и при этом считаются разными произведениями.

3.8. a) В ряд расположено n предметов. Докажите, что результат последовательного выполнения транспозиций s_1, s_2, …, s_{n-1} является циклической перестановкой в том и только том случае, когда граф, в котором множество вершин — это $[n]$, а множество ребер — это s_1, s_2, …, s_{n-1}, является деревом.

b) Докажите, что количество способов, которым циклическая перестановка множества $[n]$ может быть разложена в произведение $n-1$ транспозиций, равно T_n.

4 Парковочные функции

Вдоль улицы с односторонним движением расположено n мест для парковки автомобиля. На улицу въезжают последовательно n автомобилей с номерами от 1 до n в порядке возрастания. Каждый водитель едет к своему любимому месту парковки и, если оно не занято, припарковывается; в противном случае он едет дальше до первого свободного места и припарковывается там, если же все места дальше заняты, он уезжает (насовсем). Последовательность предпочтений, для которой все водители сумели припарковаться, называется *парковочной функцией.* Множество парковочных функций обозначим через $\mathcal{P}_n$.

4.3. Сколько существует парковочных функции $(a_1, a_2, \ldots, a_n)$, для которых соседние водители имеют разные предпочтения, т. е. $a_k \neq a_{k+1}$ при $k = 1, \ldots, n-1$?

4.4. Допустим, что на улицу, где расположено n мест для парковки, въезжают лишь $m < n$ машин. Сколько в этом случае существует последовательностей предпочтений, для которых все водители сумеют припарковаться?

4.5. Докажите, что число парковочных функций, для которых ровно k водителей $(1 \leqslant k \leqslant n)$ предпочитают парковаться на первом месте, равно $C_{n-1}^{k-1} n^{n-k}$.

4.6. Для каждой парковочной функции $a = (a_1, a_2, \ldots, a_n)$ определим разностную последовательность $c(a) = (c_1, c_2, \ldots, c_{n-1})$ по правилу

$$c_i = a_{i+1} - a_i \pmod{n+1}.$$

Сопоставим парковочной функции a помеченное дерево $t(a)$ с $n+1$ вершинами, которое задается кодом Прюфера $c(a)$.

Докажите, что описанное соответствие является биекцией между $\mathcal{P}_n$ и $\mathcal{T}_{n+1}$.

4.7. Докажите, что $$(n+1)^{n-1} = \sum_{\substack{0 \leqslant k_n \leqslant 1 \\ 0 \leqslant k_{n-1}+k_n \leqslant 2 \\ 0 \leqslant k_{n-2}+k_{n-1}+k_n \leqslant 3 \\ \vdots \\ 0 \leqslant k_2+k_3+\ldots+k_{n-1}+k_n \leqslant n-1}} \frac{n!}{(n-k_2-k_3-\ldots-k_n)!k_2!k_3!\ldots k_n!}.$$

4.8. Будем называть парковочную функцию $a = (a_1, \ldots, a_n)$ *надежной*, если при всех j, $1 \leqslant j \leqslant n-1$, по крайней мере $j+1$ водителей предпочитают припарковаться на первых j местах. Докажите, что количество надежных парковочных функций для парковки длины n равно $(n-1)^{n-1}$.

4.9. Докажите рекуррентное соотношения с помощью комбинаторных рассуждений с парковочными функциями: $P_{n+1} = \sum\limits_{k=0}^{n} C_{n+1}^{k} P_k (n-k)^{n-k}$.

5 *Инверсии на деревьях и неудобства парковочных функций*

5.1. Докажите рекуррентные соотношения с помощью комбинаторных рассуждений:

a) $T_n = \sum_{k=0}^{n-2} C_{n-2}^k (k+1) T_{k+1} T_{n-k-1}$; b) $P_{n+1} = \sum_{k=0}^{n} C_n^k (k+1) P_k P_{n-k}$;

Возьмем помеченное дерево T с $n+1$ вершиной. Вершину $n+1$ назначим корнем и введем на ребрах дерева направление от периферии к корню. Мы «считаем правильным», когда метки вершин растут при приближении к корню. Будем говорить, что вершины i и j образуют *инверсию*, $1 \leqslant i < j \leqslant n$, если вершина i лежит на пути, ведущем от вершины j к корню. Переберем все пары вершин и обозначим через $\mathrm{inv}(T)$ суммарное количество инверсий в дереве T. Максимальное возможное значение величины $\mathrm{inv}(T)$ равно $\frac{n(n-1)}{2}$, оно достигается, когда T — это путь из n звеньев и нумерацией вершин $n+1$, 1, 2, ..., n.

Пусть $a = (a_1, a_2, \ldots, a_n)$ — парковочная функция. Пусть в результате парковки первый водитель припарковался на p_1-м месте, второй — на p_2-м месте и т. д. Назовем величину

$$D(a) = \sum_{i=1}^{n} (p_i - a_i) = \frac{n(n+1)}{2} - \sum_{i=1}^{n} a_i.$$

неудобством парковочной функции a. Самое большое неудобство имеет функция $(1, 1, \ldots, 1)$, оно равно $\frac{n(n-1)}{2}$.

Теорема. Пусть n — произвольное натуральное число. Тогда при всех k, $0 \leqslant k \leqslant \frac{n(n-1)}{2}$, количество корневых помеченных деревьев на $n+1$ вершине с корнем $n+1$, имеющих k инверсий, равно количеству парковочных функций с неудобством k на улице с n парковочными местами.

Введем многочлены $F_n(x)$ и $H_n(x)$, «нумерующие» инверсии и неудобства:

$$F_0(x) = 1, \quad F_n(x) = \sum_{T \in \mathcal{T}_{n+1}} x^{\mathrm{inv}(T)}; \qquad H_0(x) = 1, \quad H_n(x) = \sum_{a \in \mathcal{P}_n} x^{D(a)},$$

Будем называть их *нумератор инверсий* и *нумератор неудобств*.

5.2. Пусть $\mathcal{T}^*$ — множество помеченных корневых деревьев на $n+3$ вершинах, у которых корень имеет степень 2 и помечен числом $n+3$, а два сына корневой вершины помечены числами $n+1$ и $n+2$. Выразите нумератор инверсий множества $\mathcal{T}^*$

$$F_n^*(x) = \sum_{T \in \mathcal{T}^*} x^{\mathrm{inv}(T)}$$

через многочлены $F_i(x)$.

5.3. Докажите, что многочлены $F_n(x)$ и $H_n(x)$ удовлетворяют при $n \geqslant 0$ одинаковым рекуррентным соотношеням

$$F_{n+1}(x) = \sum_{k=0}^{n} C_n^k (x^k + x^{k-1} + \ldots + 1) F_k(x) F_{n-k}(x). \tag{a}$$

$$H_{n+1}(x) = \sum_{k=0}^{n} C_n^k (x^k + x^{k-1} + \ldots + 1) H_k(x) H_{n-k}(x). \tag{b}$$

Теорема об инверсиях и неудобствах сразу следует из предыдущих двух задач. Но, возможно, вы сможете предложить биективное доказательство.

5.4. Докажите теорему об инверсиях и неудобствах, построив взаимно однозначное соответствие между множествами $\mathcal{T}_{n+1}$ и $\mathcal{P}_n$, при котором дереву с j инверсиями ставится в соответствие парковочная функция с неудобством j.

6 После промежуточного финиша

Помеченным плоским деревом будем называть корневое помеченное дерево, у которого сыновья любой вершины линейно упорядочены. Обозначим через $\mathcal{P}_n^k$ множество лесов, состоящих из k помеченных плоских деревьев на множестве $[n]$ с корнями 1, 2, …, k, у которых вершина n находится в дереве с корнем 1.

1.9. Докажите, что при $2 \leqslant k \leqslant n-1$ выполняется рекуррентное соотношение

$$P_n^{k-1} = (2n-k)P_n^k.$$

3.9. Пусть n и k — натуральные числа, $2 \leqslant k \leqslant n$. Найдите количество помеченных деревьев на $[n]$, у которых первая и вторая вершины соединены путем длины $k-1$.

3.10. Докажите, что количество остовных лесов в полном двудольном помеченном графе $K_{r,s}$, состоящих из $k+\ell$ корневых деревьев с корнями $1, 2, \ldots, k$ и $r+1, r+2, \ldots, r+\ell$, равно $(r\ell + sk - k\ell)r^{s-\ell-1}s^{r-k-1}$.

3.11. Докажите, что количество остовных деревьев в полном трехдольном помеченном графе $K_{r,s,t}$ с долями $V_1 = [r]$, $V_2 = \{r+1, \ldots, r+s\}$ и $V_3 = \{r+s+1, \ldots, r+s+t\}$ равно $(r+s+t)(r+s)^{t-1}(s+t)^{r-1}(t+r)^{s-1}$.

3.12. Чему равно количество плоских помеченных корневых деревьев на $n+1$ вершине?

Тетраэдральное дерево определяется по индукции аналогично треугольному. Простейшее тетраэдральное дерево (вырожденное) — это треугольник (полный граф на трех вершинах), следующее по сложности — тетраэдр (полный граф на четырех вершинах). Если уже задано тетраэдральное дерево, мы можем добавить к нему новую вершину, взяв любую треугольную грань любого из тетраэдров и построив на этой грани как на основании тетраэдр с новой вершиной (рис. 4). Формально это значит, что мы добавили к графу одну новую вершину и три новых ребра (и, если хотите, три новые треугольные грани). Заметим, что в трехмерном пространстве тетраэдры могут пересекаться, подобно тому как в треугольных деревьях, изображаемых на плоскости, могли пересекаться треугольники.

3.13. Сколько существует помеченных тетраэдральных деревьев с n вершинами?

4.10. Докажите, что $$(n-1)^{n-1} = \sum_{\substack{0\leqslant k_{n-1}\leqslant 1\\ 0\leqslant k_{n-2}+k_{n-1}\leqslant 2\\ 0\leqslant k_{n-3}+k_{n-2}+k_{n-1}\leqslant 3\\ \ldots\\ 0\leqslant k_2+k_3+\ldots+k_{n-2}+k_{n-1}\leqslant n-2}} \frac{n!}{(n-k_2-k_3-\ldots-k_{n-1})!k_2!k_3!\ldots k_{n-1}!}.$$

4.11. Докажите, что $$n^n = \sum_{\substack{0\leqslant k_1\leqslant 1\\ 0\leqslant k_1+k_2\leqslant 2\\ 0\leqslant k_1+k_2+k_3\leqslant 3\\ \ldots\\ 0\leqslant k_1+k_2+\ldots+k_{n-1}\leqslant n-1}} \frac{n!}{k_1!k_2!\ldots k_{n-1}!}.$$

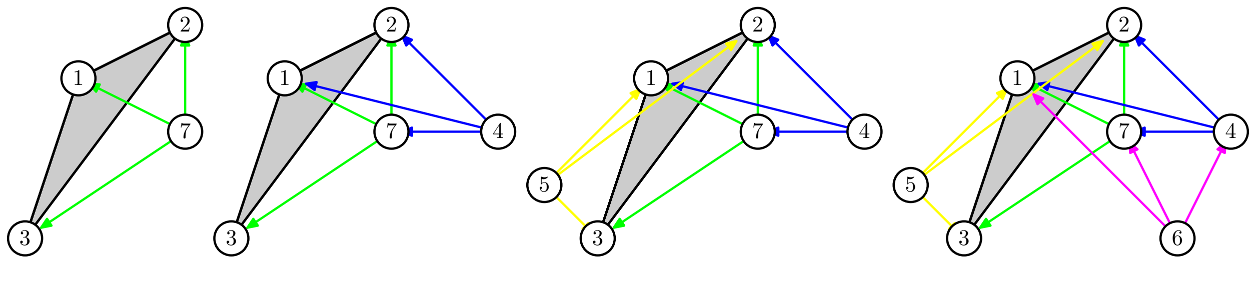


Рис. 4. Построение тетраэдрального дерева. К основному треугольнику (1, 2, 3) последовательно добавляются вершины 7, 4, 5, 6

Решения

1 *Рекурсии, тождества, биекции*

1.1. Мы взяли эту задачу в [3]. Отметим на цикле вершину A с наибольшей меткой и ту из двух соседних с ней по циклу вершин B, где метка крупнее. Уберем из дерева ребро AB, подвесим к A лист с меткой 99, и к B — лист с меткой 100. Мы построили инъективное отображение из множества унициклических графов в множество деревьев.

1.2. Решение 1. См. [6, задача 8.5] или [1] (глава 26, первое доказательство). Рассмотрим путь, ведущий из синей вершины в красную. Он задается некоторой последовательностью элементов множества $A \subset [n]$. На вершинах этого пути растут деревья. Саму последовательность можно представлять как перестановку элементов множества A. Зададим эту перестановку с помощью циклов (на элементах множества A). Мы получили набор циклов, причем на каждой вершине каждого цикла растет дерево.

С другой стороны, отображение множества $[n]$ в себя задается ориентированным графом: из вершины i рисуем стрелку в вершину $f(i)$ (разрешены петли). У этого графа все вершины имеют исходящую степень 1, поэтому он представляет собой объединение циклов, причем на каждой вершине цикла растет дерево (точнее, наоборот: дерево «вливается» в каждую вершину цикла).

Это и есть требуемая биекция.

Решение 2. Отображение из $[n]$ в $[n]$ задается последовательностью из n чисел. Возьмем дерево из условия задачи и выпишем его код Прюфера ($n-2$ числа), после чего выпишем номер красной, а затем номер синей вершины. Получилась последовательность из n чисел. Очевидно, что это биекция.

1.3. Мы взяли это утверждение в [17, п. 3.9]. Правая часть равенства подсчитывает всевозможные отображения из $[n-1]$ в $[n]$. Любое такое отображение может быть изображено в виде ориентированного графа на n вершинах: из каждой вершины выходит одно ребро, показывающее, куда отображается эта вершина. У вершины n могут быть только входящие ребра, и компонента связности вершины n — это дерево, в котором все стрелки «показывают в сторону n». Левая часть равенства перечисляет такие графы, классифицируя их по виду дерева, содержащего вершину n.

1.4. [14, Теорема 2.1]. На корневом дереве задано направление от корня к периферии. Построим отображение из множества $\mathcal{F}_n^{k-1}$ в $\mathcal{F}_n^k$. Для этого возьмем лес с $k-1$ деревом, найдем в нем вершину k, отломаем ее и все, что на ней растет, и посадим в качестве отдельного корневого дерева. Как правило получится дерево из $\mathcal{F}_n^k$ за исключением одного случая: если мы отломали ветку от первого дерева и вершина n находится на отломанной ветке. В этом случае сделаем коррекцию: поменяем метки 1 и k местами.

Посмотрим, сколько прообразов имеет при таком отображении произвольный лес из $\mathcal{F}_n^k$. Чтобы построить прообраз леса, нужно взять в нем k-е дерево и прицепить в качестве ветки к любой вершине первого, второго, ..., $(k-1)$-го дерева — это если мы подразумеваем, что не было коррекции. А для прообразов с коррекцией нужно первое дерево в качестве ветки прицепить к любой вершине k-го дерева и поменять метки 1 и k. Итого, каждая из n вершин имеющегося графа может быть использована для построения (уникального) прообраза. Таким образом, $F_n^{k-1} = nF_n^k$.

1.5. Ответ: $F_{r,s}^{k-1} = sF_{r,s}^k$ при $2 \leqslant k \leqslant r$. [14, Следствие 3.1]. Рекурсия строится так же, как в задаче 1.4. Поскольку вершина с номером k лежит в множестве V_1, при построении обратного отображения поддерево с корнем k может быть подвешено к s вершинам.

1.6. а) Это результат из [18]. Обозначим через E_n количество помеченных деревьев с вершинами на $[n]$, содержащих определенное ребро, например, ребро 1–2. Подсчитывая все ребра во всех деревьях, приходим к тождеству

$$\frac{n(n-1)}{2}E_n = (n-1)T_n.$$

Таким образом, $T_n = \frac{1}{2}nE_n$. Теперь, чтобы найти E_n, достаточно выбрать, сколько вершин содержит дерево, висящее на вершине 1, а сколько на вершине 2, и какие именно это деревья. Получаем формулу $T_n = \frac{n}{2}\sum\limits_{k=0}^{n-2} C_{n-2}^k T_{k+1}T_{n-k-1}$.

b) Тождество из этой задачи выводится из тождества задачи 5.1 a) с помощью «хрестоматийного метода Гаусса»: нужно сложить два экземпляра суммы из задачи 5.1 a), поменять в одном из них индекс суммирования k на $n-2-k$ и сложить эти суммы почленно.

Выведем тождество задачи 1.3 из тождества 5.1 a), для этого запишем отдельно слагаемое для $j = n$, а в оставшейся сумме введем индекс суммирования k, где $j = k+1$:

$$n^{n-2} + \sum_{k=0}^{n-2} C_{n-1}^k (k+1)^{k+1}(n-k-1)^{n-k-1} = n^{n-1}.$$

Перенесем слагаемое n^{n-2} в правую часть, а в левой части преобразуем биномиальный коэффициент по формуле $C_{n-1}^k(n-k-1) = (n-1)C_{n-2}^k$:

$$\sum_{k=0}^{n-2}(n-1)C_{n-2}^k(n-k-1)(k+1)^{k+1}(n-k-1)^{n-k-3} = n^{n-2}(n-1).$$

Сократив на $(n-1)$, получаем формулу из задачи 5.1 a) с точностью до замены индекса суммирования.

1.7. [4] Пусть $A \in \mathcal{T}(n, k-1)$ — одно из деревьев, v — любая из его $n-k$ вершин, несмежных с вершиной 1. Заменим ребро, ведущее из вершины v к предку, на ребро, ведущее в корень. Полученное дерево обозначим B (очевидно $B \in \mathcal{T}(n,k)$), а пару деревьев (A, B) назовем *связкой*.

Подсчитаем двумя способами количество связок. С одной стороны, число связок равно $(n-k)T(n, k-1)$, поскольку связка однозначно определяется деревом A и вершиной v, дерево можно выбрать $T(n, k-1)$ способами, после чего вершина v выбирается $(n-k)$ способами. С другой стороны, связку можно получить, если мы выберем $T(n, k)$ способами дерево B, после чего удалим одно из ребер, выходящих из корня, и образовавшуюся «отломанную ветку» соединим с любой некорневой вершиной оставшегося дерева — всего получится

$$(n-1-n_1) + (n-1-n_2) + \ldots + (n-1-n_k) = (n-1)(k-1)$$

вариантов, здесь через n_i обозначено число вершин в «отломанной ветке», образовавшейся при удалении i-го ребра, выходящего из корня. Итого $(n-1)(k-1)T(n, k)$ вариантов. Значит, $(n-1)(k-1)T(n, k) = (n-k)T(n, k-1)$.

1.8. Ответ: $(n-k-2)\Delta(n, k) = k(2n-4)\Delta(n, k+1)$. Это утверждение из [9].

Отметим, что построение треугольного дерева, описанное в определении, можно начинать с любого ребра: возьмем любое ребро, добавим к нему треугольники, которые его содержат, потом добавим треугольники, примыкающие к уже построенным ребрам и т. д. Более того, пусть начальное ребро фиксировано, а при добавлении к дереву очередной вершины C, мы рисуем на ребрах CA, CB стрелки, выходящие из вершины C. Тогда ориентированный граф, изображающий треугольное дерево, не зависит от порядка, в котором добавлялись вершины!

Рассмотрим треугольное дерево G, у которого корневое ребро uv принадлежит k треугольникам, обозначим их uvw_1, uvw_2, $\ldots$, uvw_k, и построим по нему дерево с $k+1$ треугольниками (рис. 5). Возьмем произвольную вершину w, не совпадающую ни с одной из вершин w_i, и рассмотрим минимальное треугольное поддерево, содержащее вершины u, v

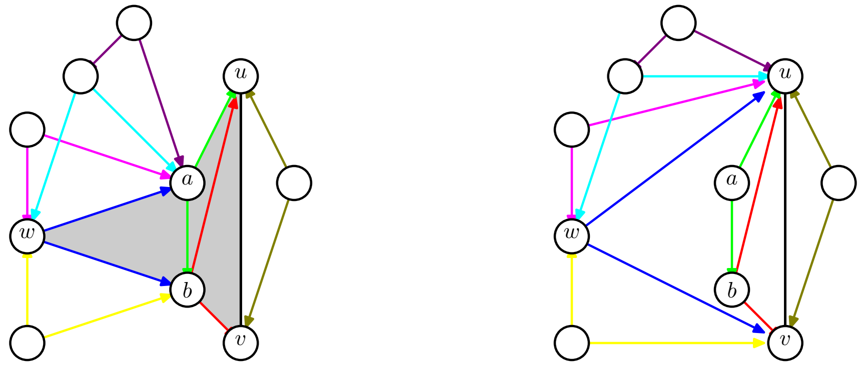


Рис. 5. Слева дерево с k треугольниками на корневом ребре uv, справа с $k+1$ треугольниками ($k = 1$). Минимальное дерево, содержащее вершины w, u и v, закрашено серым цветом. От каждой вершины, кроме вершин u и v, ведут две стрелки к ребру, на которое был повешен треугольник, соответствующий этой вершине

и w. В силу минимальности из вершины w в этом поддереве выходят два ребра. Уберем эти ребра и вместо них проведем ребра wu и wv. В результате у корневого ребра появится еще один треугольник. Части дерева G, которые висели на убранных ребрах, перевесим на новые ребра wu и wv. (При выполнении этих действий могло случиться, что мы убрали ребро, соединяющее w с u или v, и тут же добавили его снова.) Выбрать вершину w можно $n-k-2$ способами, поэтому мы имеем $(n-k-2)\Delta(n,k)$ вариантов выполнения этой конструкции.

Теперь опишем обратную операцию: по дереву, у которого корневое ребро uv принадлежит $k+1$ треугольникам, построим дерево, у которого корневое ребро uv принадлежит k треугольникам. Для этого возьмем одну из $k+1$ вершин треугольников, связанных с корнем, пусть это будет вершина w, и «пересадим» треугольник uvw с ребра uv на какое-нибудь другое ребро ab. Это делается следующим образом: пусть G_1 и G_2 — треугольные поддеревья, растущие на ребрах wu и wv. Выберем произвольное ребро ab, не лежащее в $G_1 \cup G_2$ (и не совпадающее с uv), и заменим ребра wu и wv на ребра wa и wb. При этом поддеревья G_1 и G_2, висевшие на ребрах wu и wv, перевесим на ребра wa и wb соответственно.

Подсчитаем, сколькими способами может быть выполнена такая конструкция. В качестве ab может выбираться любое из $2n-4$ некорневых ребер, причем, если ребро ab принадлежит треугольному поддереву, висящему на корневом треугольнике uvw', то в качестве треугольника uvw, который мы перевешиваем на это ребро, можно взять любой из k корневых треугольников uvw, где $w \neq w'$. Итого $k(2n-4)\Delta(n,k+1)$ способов.

1.9. [14, Следствие 4.1]. Попробуем построить такую же рекурсию, как в задаче 1.4. На этот раз мест, куда можно подвесить поддерево с корнем k, больше. Для каждой вершины оно равно числу ее потомков плюс один. Суммарно по всем вершинам получается число всех ребер плюс число всех вершин. А число ребер в лесу из k деревьев равно $n-k$.

1.10. Ответ: $P^{r-1}_{n,p} = (p+1)P^{r}_{n+1,p+1}$. [14, Теорема 4.3]. Рекурсия почти как в задаче 1.4. В качестве r-го дерева берем поддерево с корнем в вершине r. На то место, где была вершина r, подвесим лист. Заметим, что если бы мы этого не сделали, то пронумерованный предок вершины r мог оказаться листом, если у него не было больше потомков. Число листьев и общее число вершин n увеличились на 1. Если теперь вершина $n-p$ оказалась в дереве с корнем r, переставим метки 1 и r. Обратно, убираем любой из $p+1$ листьев и вместо него подвешиваем дерево с корнем r если лист находится в другом дереве или подвешиваем дерево с корнем 1 и переставляем метки 1 и r, если лист находится в r-м дереве.

2 *Код Прюфера*

2.1. Эта задача 2.2.4 из книжки [3] — упражнение на понимание того, что представляет собой код Прюфера.

2.2. Ответ: $(n-1)^{n-1}$.

Код Прюфера некорневого дерева, у которого вершина n является листом, не содержит символа «n». Потому имеется $(n-1)^{n-2}$ таких кодов. Выбор корня увеличивает число вариантов в $(n-1)$ раз.

2.3. Ответ: $C^9_{n-2}(n-1)^{n-11}$. Потому что это число равно количеству заполнений n ячеек $n-2$ различными предметами, для которых первая ячейка содержит 9 предметов.

2.4. Ответ: $\frac{1}{2}(n-1)(n-2)\dots(n-k+1)n^{n-k}$.

Действительно, количество способов выбрать цикл равно $\frac{n(n-1)(n-2)\dots(n-k+1)}{2k}$ — выбираем последовательно первую, вторую и т. д. вершины цикла, после чего учтем, что нам не важно, которая вершина в цикле числится первой и каково направление цикла. Осталось подсчитать количество унициклических графов, содержащих цикл $(n, n-1, \dots, n-k+1)$. Уберем ребро, соединяющее n и $n-k+1$. Количество возможных кодов Прюфера для получившегося дерева равно kn^{n-1-k}.

Действительно, после $n-k$ шагов выписывания кода Прюфера нашего дерева от него останется как раз путь n, $n-1$, …, $n-k+1$. Первые $n-k-1$ элементов кода Прюфера произвольны, $(n-k)$-й элемент — это число от $n-k+1$ до n, а все остальные элементы детерминированы.

Мы взяли эту задачу в [3].

2.5. Мы взяли это утверждение в [5]. Если присмотреться, оно есть и в [7]. В свете кода Прюфера, кажется, это почти тавтология.

2.6. [7]. По алгоритму Прюфера каждому такому дереву взаимно однозначно соответствует код Прюфера — одночлен $x_{i_1}x_{i_2}\dots x_{i_{n-2}}$, который после «упрощения» приводится к виду $x_1^{k_1}x_2^{k_2}\dots x_n^{k_n}$ (где $k_1+k_2+\dots+k_n=n-2$). Таким образом количество разных кодов равно полиномиальному коэффициенту $\frac{(n-2)!}{k_1!k_2!\dots k_n!}$.

3 Результаты

3.1. Мы взяли это утверждение в [17]. Для каждого дерева с помеченными вершинами назначим вершину n корнем, тем самым мы зададим направление от корня и, как следствие, отношение «предок–потомок». Теперь каждому помеченному дереву с n вершинами поставим в соответствие реберно помеченное дерево с n вершинами. Для этого поставим на каждом ребре ij, где i потомок j, метку i.

Полученное реберно помеченное дерево позволяет восстановить исходную разметку вершин, если мы укажем, которая из вершин была корнем и присвоим ей метку n. Таким образом, каждое реберно помеченное дерево могло получиться при этом отображении из n различных вершинно помеченных деревьев.

3.2. Мы взяли эту задачу в [3]. Будем считать число корневых деревьев. Нарисуем дерево на плоскости, у каждого ребра зададим направление от корня. Теперь сопоставим дереву код: начиная от корня, будем обходить дерево, как будто это система стен на плоскости. Когда движемся по стрелке — пишем 1, а против — 0. Получим последовательность из $2n-2$ нулей и единиц. Нетрудно понять, что каждой последовательности соответствует не более одного дерева.

3.3. a) Теорема Кэли сразу следует из задачи 1.4 и равенства $F_n^{n-1} = 1$.

c) А также из задачи 1.7 и равенства $T(n,k) = C_{n-2}^{k-1}(n-1)^{n-k-1}$. Действительно, количество помеченных деревьев равно сумме

$$\sum_{k=1}^{n-1} T(n,k) = \sum_{k=1}^{n-1} C_{n-2}^{k-1}(n-1)^{n-k-1} = \sum_{k=0}^{n-2} C_{n-2}^{k}(n-1)^{n-2-k} = ((n-1)+1)^{n-2}.$$

b) Что касается задачи 1.6, теорема Кэли выводится из нее в [18] с помощью производящих функций и формулы обращения Лагранжа. Но, возможно, есть и элементарный метод.

d) Из задачи 2.6 теорема Кэли выводится с помощью суммирования в духе полиномиальной теоремы [7]:

$$\sum_{\substack{k_1+k_2+\ldots+k_n=n-2\\ k_i\geqslant 0}} \frac{(n-2)!}{k_1!k_2!\ldots k_n!} = n^{n-2}.$$

3.4. [14, Следствие 2.3]. Существует всего один лес с $n-1$ деревьями. Применяя рекурсивное равенство $F_n^{k-1} = nF_n^k$ из задачи 1.4 несколько раз, получим, что $F_n^k = n^{n-k-1}$. Число лесов, в которых вершина с номером n содержится не в первом, а в любом другом фиксированном дереве, равно F_n^k. Следовательно, число лесов без условия на вершину n равно kn^{n-k-1}. Число способов выбрать корни для k деревьев равно C_n^k, отсюда искомое количество лесов равно $C_n^k kn^{n-k-1} = C_{n-1}^{k-1}n^{n-k}$.

3.5. Мы взяли это утверждение в [17, п. 4.3].

3.6. Р е ш е н и е 1. [14, Следствие 3.1]. Будем подсчитывать не только число деревьев, но и число лесов (у которых вершина $r+1$ находится в дереве с корнем 1). Воспользуемся рекурсивной формулой из задачи 1.5:

$$F_{r,s}^{k-1} = sF_{r,s}^k \quad \text{при} \quad 2 \leqslant k \leqslant r.$$

Пользуясь этой формулой, сведем вопрос к подсчету числа лесов с r деревьями. Это число лесов равно $F_{r,s}^r = r^{s-1}$, так как число способов выбрать среди r корней предков для $s-1$ вершин равно r^{s-1}.

Р е ш е н и е 2 (код Прюфера). Код Прюфера такого такого дерева содержит $r-1$ предков одной доли и $s-1$ предков другой доли, так как в конце всегда остаются две вершины разных долей. Каждый из $r-1$ предков может быть выбран s способами и аналогично, каждый из $s-1$ предков может быть выбран r способами.

3.7. а) Решение 1. Утверждение задачи взято в [9], где оно доказано пятью способами. Мы приводим третье доказательство из [9].

Воспользуемся рекурсией из решения 1.8 и тем, что $\Delta(n,k)=1$ при $k=n-2$. Получаем

$$\Delta(n,k)=(2n-4)\frac{k}{n-2-k}\Delta(n,k+1)=(2n-4)^2\frac{k(k+1)}{(n-2-k)(n-2-(k+1))}\Delta(n,k+2)=$$
$$=\ldots=(2n-4)^{n-2-k}\frac{k(k+1)\ldots(n-3)}{(n-2-k)(n-2-(k+1))\cdots 1}\Delta(n,n-2)=$$
$$=(2n-4)^{n-2-k}\frac{(n-3)!}{(k-1)!(n-3-(k-1))!}\Delta(n,n-2)=(2n-4)^{n-2-k}C_{n-3}^{k-1}.$$

Суммируя по всем возможным k и применяя формулу бинома Ньютона, вычисляем количество корневых треугольных деревьев: $\sum\limits_{k=1}^{n-2}C_{n-3}^{k-1}(2n-4)^{n-k-2}=(2n-3)^{n-3}$.

Решение 2 (код Прюфера). Закодируем помеченное корневое треугольное дерево кодом, аналогичным коду Прюфера.

Построение кода по дереву. Пусть дано треугольное помеченное корневое дерево с корневым ребром 1–2. Сначала по заданной разметке вершин дерева определим специальную разметку ребер. Для этого поставим на ребрах стрелки, как это описано в начале решения задачи 3.7. Тогда из каждой вершины, не принадлежащей корневому ребру, выходит две стрелки. Если вершина имеет метку x, то пометим выходящие из нее ребра xa и xb, где xa ведет в вершину с меньшей меткой, а xb — с большей. Корневое ребро будем обозначать 1–2. Таким образом, наше дерево содержит $2n-3$ ребра с «унифицированными» названиями:

$$12,\ 3a,\ 3b,\ 4a,\ 4b,\ \ldots,\ na,\ nb.$$

Запишем код, пользуясь новыми названиями ребер. Листьями треугольного дерева будем называть вершины степени 2. На каждом шаге смотрим на все вершины, являющиеся листьями текущего треугольного дерева, выбираем из них вершину с *наибольшим* номером, отрываем ее от дерева и записываем название ее ребра-предка. На последнем шаге отрываем вершину с оставшимся номером от корневого ребра 1–2, его можно не записывать.

Восстановление дерева по коду. Листьями дерева являются вершины с номером x, если оба ребра xa и xb не содержатся в коде, также листьями не могут быть вершины 1 и 2. Выпишем такие номера вершин. В этом списке вершина с наибольшим номером была оторвана первой, значит, первое ребро в коде — это ее ребро-предок. Записываем, какое ребро является предком для этой вершины, уберем ее из списка листьев, а ребро из кода. После удаления ребра из кода, могут появиться новые вершины, которые теперь следует считать листьями, добавим их в список листьев. Снова найдем в этом списке вершину с наибольшим номером и т. д.

В результате выполнения этого алгоритма, мы получим последовательность вершин с их ребрами-предками. В отличие от обычного дерева треугольное трудно восстановить, просматривая эту последовательность с начала, так как неизвестен один из номеров вершин ребра xa или xb. Зато, начиная с хвоста, это делается легко. Рисуем корневое ребро 1–2. Первой к нему была присоединена вершина, которой нет в последовательности. На следующем шаге присоединяется вершина y к ребру xa или xb, а эти ребра уже нарисованы, следовательно, известны обе их вершины.

Почему в списке всегда имеется хотя бы один лист? В текущем коде $n-3-k$ позиции, где k — число уже рассмотренных позиций. А число вершин, могущих оказаться листьями, равно $n-2-k$ (слагаемое 2 присутствует, так как вершины 1 и 2 не листья), то есть всегда на единицу больше. Даже если на всех позициях кода стоят ребра, соответствующие различным вершинам, в списке листьев есть один элемент.

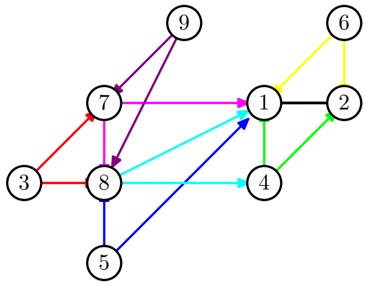


Рис. 6. Пример построения кода треугольного дерева (7b)(12)(8a)(7b)(8a)(4a).

Код содержит $n-3$ позиции. Всего ребер $2n-3$. На каждой позиции может быть любое ребро. Значит, число таких кодов равно $(2n-3)^{n-3}$. Поскольку каждому коду соответствует ровно одно дерево, а по каждому дереву строится код, корневых треугольных деревьев столько же.

Пример. Пусть имеется 8 вершин и код Прюфера (7b, 1–2, 8a, 7b, 8a, 4a).

Номер шага	Текущий код	Список листьев	Подвешенная вершина
1	(7b, 1–2, 8a, 7b, 8a, 4a)	3, 5, 6, 9	9
2	(1–2, 8a, 7b, 8a, 4a)	3, 5, 6	6
3	(8a, 7b, 8a, 4a)	3, 5	5
4	(7b, 8a, 4a)	3	3
5	(8a, 4a)	7	7
6	(4a)	8	8

Не перечислена вершина 4, значит, она была подвешена к ребру 1–2 первой. Нарисуем треугольник на вершинах 1, 2, 4. Далее были подвешены вершины 8, 7, 3, 5, 6, 9 к ребрам 4a, 8a, 7b, 8a, 1–2, 7b соответственно. Подвешиваем вершину 8 к ребру 4a (т. е. к ребру 2–4) и т. д. Получившееся треугольное дерево показано на рис. 6.

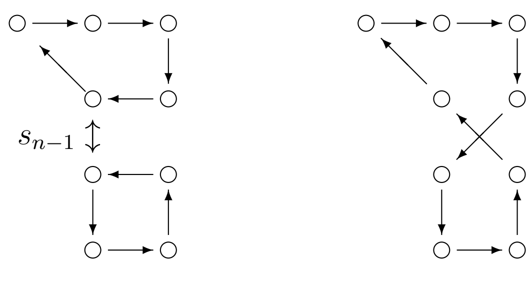
b) Так как выделенное ребро треугольного дерева может быть помечено любой парой меток, а не только 1–2, а любое из $2n-3$ ребер в треугольном дереве можно назначить корнем, выполняется соотношение

$$C_n^2\Lambda_n = (2n-3)\Delta_n.$$

3.8. Этот результат J. Dénes мы приводим по книге [11] (лемма 3.15 и теорема 3.16).

a) Если граф, изображающий транспозиции, несвязен, то перестановка не сможет перенести предмет, отнесенный к одной из компонент, на место предмета из другой компоненты. Таким образом, граф связен и в нем $n-1$ ребро. Следовательно, это дерево.

Проверим теперь, что произведение транспозиций, построенное по произвольному помеченному дереву, является циклической перестановкой множества его вершин. Это утверждение легко доказывается индукцией по числу сомножителей. База $n=1$ тривиальна. Докажем переход. Зафиксируем произведение $s_1s_2\ldots s_{n-1}$. При удалении ребра, соответствующего множителю s_{n-1}, дерево распадется на две компоненты. Оставшееся произведение $s_1s_2\ldots s_{n-2}$ переставляет по циклу элементы в каждой из компонент. Добавление множителя s_{n-1} объединяет эти два цикла в один (см. рис).



b) Зафиксируем стандартное обозначение — будем задавать перестановку двустрочной диаграммой. Запись $\left(\begin{smallmatrix} 1 & 2 & \dots & n \\ a_1 & a_2 & \dots & a_n \end{smallmatrix}\right)$ обозначает перестановку, которая переставляет первый предмет на место a_1, второй предмет — на место a_2 и т. д. Тогда, например, $C = \left(\begin{smallmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & 1 \end{smallmatrix}\right)$ это одна из циклических перестановок. При необходимости столбцы диаграммы, задающей перестановку, можно переставлять. Тогда понятно, что обратная перестановка задается диаграммой, которая получатся из диаграммы исходной перестановки перестановкой строк.

Построим биекцию между множеством помеченных деревьев на n вершинах и и множеством разложений перестановки C в произведение $n-1$ транспозиций.

Возьмем произвольное помеченное дерево T. Сначала построим по этому дереву вспомогательное произведение транспозиций. Для этого назначим вершину 1 корнем и зафиксируем на ребрах направление к корню. Для каждой вершины i обозначим выходящее из нее ребро через s_i и тем же символом будем обозначать транспозицию чисел на концах ребра. Рассмотрим циклическую перестановку $S = s_2 \dots s_n$ (она циклическая по пункту a)) Запишем S в качестве диаграммы: $S = \left(\begin{smallmatrix} 1 & a_2 & \dots & a_n \\ a_2 & a_3 & \dots & 1 \end{smallmatrix}\right)$ и рассмотрим перестановку $F = \left(\begin{smallmatrix} 1 & 2 & \dots & n \\ 1 & a_2 & \dots & a_n \end{smallmatrix}\right)$. Заметим, что

$$FSF^{-1} = C,$$

поскольку для всех k предмет, стоящий на k-м месте, перемещается под действием перестановки F на a_k-е место, потом S переставляет его на a_{k+1} место, и наконец, перестановка F^{-1} отправляет этот предмет на $(k+1)$-е место.

Рассмотрим набор транспозиций $u_i = Fs_iF^{-1}$ (в том, что перестановки u_i являются транспозициями, читатель может легко убедиться сам). Тогда

$$u_2 \dots u_n = Fs_2F^{-1}Fs_3F^{-1} \dots Fs_nF^{-1} = Fs_2s_3 \dots s_nF^{-1} = FSF^{-1} = C.$$

Таким образом, мы сопоставили произвольному дереву набор транспозиций u_i, произведение которых равно перестановке C:

Чтобы доказать, что данное сопоставление — биекция, покажем, как можно восстанавить дерево по произведению транспозиций u_i.

Пусть дано произведение $u_2 \dots u_n$, причем пусть этот набор транспозиций соответствует некоторому дереву $\widetilde{T}$, будем считать вершину 1 его корнем. Предположим на секундочку, что нам известна перестановка F. Тогда мы знаем, что $F^{-1}u_iF = s_i$ (и кстати, $F(1) = 1$, т. е. перестановка сохраняет метку корня). Таким образом, если транспозиция s_i меняет местами i и j, то транспозиция u_i меняет местами $F^{-1}(i)$ и $F^{-1}(j)$. Это значит, что если к меткам вершин дерева T применить перестановку F^{-1}, получится в точности дерево $\widetilde{T}$!

Поскольку дерево $\widetilde{T}$ нам известно, осталось восстановить перестановку F. По построению корень дерева $\widetilde{T}$ в 1. Зафиксировав направление к корню, мы для каждого i с легкостью найдем в дереве $\widetilde{T}$ ребро u_i, при этом соответствующее ребро в дереве T выходит из вершины i, следовательно, $F^{-1}(i)$ равно номеру вершины, из которой выходит ребро u_i.

3.9. Ответ: $kn^{n-k-1}(n-2)(n-3)\dots(n-k)$.

Аналогично 2.4. Существует $(n-2)(n-3)\dots(n-k)$ способов построить путь из первой вершины во вторую. Количество кодов Прюфера для дерева, содержащего такой путь, равно kn^{n-k-1}. Мы взяли это утверждение в [5, задача 5.93].

3.10. В [17, п. 3.5] это утверждение советуют доказывать с помощью рекурсии, мы воспользуемся рекурсией из задачи 1.5.

Как и в задаче 3.6, используя рекурсию из 1.5, находим, что

$$F^q_{a,b} = bF^{q+1}_{a,b} = \dots = b^{a-q}F^a_{a,b} = b^{a-q}a^{b-1},$$

где $F^q_{a,b}$ — число двудольных лесов с долями по a и b вершин, состоящих из q (где $q \leqslant a$) корневых деревьев с корнями $1, 2, \dots, q$ из первой доли, у которых вершина $a+1$ находится в дереве с корнем 1. Если мы умножим это значение на число корней q, то получим число лесов без условия на то, где находится вершина $a+1$.

Кроме этих соображений нам понадобятся биномиальные тождества

$$\sum_{\gamma=0}^{r-k} C_{r-k}^{\gamma} x^{\gamma} y^{r-k-\gamma} = (x+y)^{r-k} \quad \text{и} \quad \sum_{\gamma=0}^{r-k} C_{r-k}^{\gamma} \gamma x^{\gamma} y^{r-k-\gamma} = (r-k)x(x+y)^{r-k-1},$$

Приступим к решению задачи. Выберем в первой доле γ вершин, которые будут подвешены в качестве сыновей к ℓ корням второй доли. Подвесить γ вершин к ℓ корням можно γ^{ℓ} способами. После этого удалим ℓ корней второй доли, а выбранные вершины назначим корнями. У нас получится двудольный лес с долями по r и $s-\ell$ вершин, который состоит из $k+\gamma$ корневых деревьев с корнями, принадлежащими первой доле. Количество таких лесов равно $(k+\gamma)F_{r,s-\ell}^{k+\gamma}$. Следовательно, искомое значение равно

$$\sum_{\gamma=0}^{r-k} C_{r-k}^{\gamma} \ell^{\gamma} (\gamma+k) F_{r,s-\ell}^{k+\gamma} = \sum_{\gamma=0}^{r-k} C_{r-k}^{\gamma} \ell^{\gamma} (k+\gamma)(s-\ell)^{r-k-\gamma} r^{s-\ell-1} =$$
$$= kr^{s-\ell-1} \sum_{\gamma=0}^{r-k} C_{r-k}^{\gamma} \ell^{\gamma} (s-\ell)^{r-k-\gamma} + r^{s-\ell-1} \sum_{\gamma=0}^{r-k} C_{r-k}^{\gamma} \gamma \ell^{\gamma} (s-\ell)^{r-k-\gamma} =$$
$$= r^{s-\ell-1}(ks^{r-k} + (r-k)\ell s^{r-k-1}) = r^{s-\ell-1} s^{r-k-1}(ks + r\ell - k\ell).$$

3.11. [14, Следствие 3.3]. Пусть $V_1 = [r]$, $V_2 = \{r+1, \dots, r+s\}$, $V_3 = \{r+s+1, \dots, r+s+t\}$ и $V = V_1 \cup V_2 \cup V_3$. Обозначим через $F_{r,s,t}^{k}$ количество корневых трехдольных лесов на множестве вершин V, состоящих из k деревьев с корнями $1, 2, \dots, k$, в которых вершина с номером $r+1$ является потомком вершины 1, и через $\bar{F}_{r,s,t}^{k}$ количество корневых трехдольных лесов на множестве вершин V с корнями $2, 3, \dots, k+1$, в которых вершина с номером 1 является потомком вершины $r+1$. Нас интересует количество деревьев, т. е. $F_{r,s,t}^{1}$.

$$F_{r,s,t}^{1} \underset{(2)}{=} (s+t)^{r-1} F_{r,s,t}^{r} \underset{(3)}{=} (s+t)^{r-1} \bar{F}_{r,s,t}^{r} \underset{(4)}{=}$$
$$\underset{(4)}{=} (s+t)^{r-1}(r+t)^{s-1} \bar{F}_{r,s,t}^{r+s-1} \underset{(5)}{=} (s+t)^{r-1}(r+s+t)(r+s)^{t-1}.$$

(2) аналогично задаче 1.5 при $2 \leqslant k \leqslant r$ получаем рекурсивную формулу

$$F_{r,s,t}^{k-1} = (s+t) F_{r,s,t}^{k}$$

и, пользуясь этой формулой несколько раз, переходим к лесу с r деревьями;

(3) используем равенство $F_{r,s,t}^{r} = \bar{F}_{r,s,t}^{r}$ (леса в этих множествах отличаются только выбором корня для дерева, содержащего вершину 1);

(4) при $r+1 \leqslant k \leqslant r+s-1$ получаем рекурсивную формулу

$$\bar{F}_{r,s,t}^{k-1} = (r+t) \bar{F}_{r,s,t}^{k}$$

снова аналогично задаче 1.5 и переходим к лесу с $r+s-1$ деревьями; Переход от лесов с корнями $1, 2, \dots, r$ к лесам с корнями $2, 3, \dots, r+1$ обусловлен тем, что в этом рекурсивном равенстве вершина с номером k должна лежать во второй доле.

(5) число лесов с большим количеством деревьев легко считается:

$$\bar{F}_{r,s,t}^{r+s-1} = (r+s)^{t} + t(r+s)^{t-1},$$

в первом слагаемом вершина с номером 1 является сыном вершины с номером $r+1$, а оставшиеся t вершин третьей доле можно соединить с любыми $r+s$ вершин других долей, во втором слагаемом вершина с номером 1 является сыном некоторой вершины из третьей доли, а оставшиеся $t-1$ вершин аналогично распределяются по $r+s$ местам.

3.12. Ответ: $(2n)!/n!$. Тем самым количество плоских помеченных некорневых деревьев на $n+1$ вершине равно числу Каталана $\frac{1}{n+1}C_{2n}^n$. [14, Следствие 4.2].

3.13. Ответ: $C_n^3(3n-8)^{n-5}$ из [17] (решение там не приводится).

a) Число корневых тетраэдральных деревьев может быть посчитано с помощью кодирования, аналогично числу корневых треугольных деревьев.

Построение кода по дереву. Выберем корневой треугольник, например, треугольник $(1,2,3)$. Элементами кодовой последовательности будут треугольники — грани тетраэдров. Пусть вершина x была подвешена к грани с номерами вершин a, b и c, где $a<b<c$, будем называть эту грань предком вершины x. Обозначим грань (x,a,b) через $x\alpha$, (x,a,c) через $x\beta$, (x,b,c) через $x\gamma$. Таким образом, грани тетраэдра получили обозначения $x\alpha$, $x\beta$, $x\gamma$, где $x=4,\ \dots,\ n$, а корневую крань мы обозначим $(1,2,3)$. Всего граней $3n-8$. Назовем листом тетраэдрального дерева вершину степени 3. На каждом шаге отрываем лист с наибольшим номером и записываем его грань-предка. Когда остается последний тетраэдр (4 вершины), мы останавливаемся.

Построение дерева по коду. Листьями являются вершины тетраэдального дерева с такими номерами $x\in\{4,\dots,n\}$, для которых текущий код не содержит ни одной из граней $x\alpha$, $x\beta$, $x\gamma$.

Пример. Пусть имеется 7 вершин и код Прюфера $(4\beta,\ (1,2,3),\ 7\alpha)$.

Номер шага	Текущий код	Список листьев	Подвешенная вершина
1	$(4\beta,\ (1,2,3),\ 7\alpha)$	6, 5	6
2	$((1,2,3),\ 7\alpha)$	5, 4	5
3	(7α)	4	4
4	$()$	7	7

Читаем список подвешенных вершин в обратном порядке и получаем, что последовательно были подвешены вершины 7 к грани $(1,2,3)$, 4 к грани $7\alpha=(7,1,2)$, 5 к грани $(1,2,3)$, 6 к грани $4\beta=(4,1,7)$.

Получившееся тетраэдральное дерево показано на рис. 4.

Код содержит $n-4$ грани. Значит, всего возможных кодов $(3n-8)^{n-4}$.

b) Так как выделенное ребро тетраэдрального дерева может быть помечено любой тройкой меток, а не только 1, 2, 3, а любой из $5n-8$ треугольников в тетраэдральном дереве можно назначить корнем, выполняется соотношение

$$C_n^3\Lambda_n=(3n-8)\Delta_n.$$

4 Парковочные функции

4.1. Эта знаменитая задача поставлена в [15].

Увеличим стоянку, добавив $(n+1)$-е место для парковки, и замкнем улицу в цикл так, чтобы с $(n+1)$-го места она вела обратно к первому. Теперь имеется ровно $(n+1)^n$ последовательностей предпочтения (каждый из n водителей независимо от других может предпочитать одно из $n+1$ мест). Приезжая на круговую стоянку по описанному правилу, все n водителей смогут припарковаться при любых предпочтениях, причем одно место парковки останется свободным. Последовательность предпочтений удовлетворяет требованию исходной задачи тогда и только тогда, когда свободным остается $(n+1)$-е место. Действительно, то, что $(n+1)$-е место остается свободным, означает, что оно не является любимым ни для кого из водителей, и никто из водителей не проезжает мимо него в поисках парковки (иначе он припарковался бы там, а не проехал мимо). В таком случае можно закрыть $(n+1)$-е место парковки и прилегающий участок дороги (то есть вернуться к исходной задаче), не нарушая процесса поиска парковок.

Разобьем все последовательности предпочтений круговой стоянки на $(n+1)^{n-1}$ групп по $n+1$ последовательностей в каждой: в одну группу с последовательностью $(a_1,\dots,a_n)$ отнесем те, которые получаются из нее циклическим сдвигом номеров мест парковки, т. е. $(a_1+1,\dots,a_n+1)$, $(a_1+2,\dots,a_n+2)$, …, $(a_1+n,\dots,a_n+n)$ (сложения по модулю $n+1$). В каждой группе имеется ровно одна последовательность предпочтений, удовлетворяющая требованию исходной задачи, а именно, та, в которой свободному месту соответствует номер $n+1$. Значит, искомое число последовательностей равно числу групп, т. е. $(n+1)^{n-1}$.

4.2. Более-менее ясно, что происходит, если поменять местами два соседних автомобиля.

4.3. Ответ: n^{n-1}.

Действуя как и в задаче 4.1, получаем, что на расширенной парковочной улице количество всевозможных последовательностей предпочтения равно $(n+1)n^{n-1}$, поскольку у первого водителя может быть $n+1$ любимое место, а у всех остальных — по n. При этом в каждой группе из $n+1$ последовательностей, отличающихся циклическим сдвигом, содержится ровно одна парковочная функция. Мы взяли эту задачу из книги [12].

4.4. Ответ: $(n+1-m)(n+1)^{m-1}$ последовательностей.

Решение 1. Рассмотрим расширенную парковочную улицу, как в задаче 4.1. Для m водителей существует $(n+1)^m$ всевозможных «расширенных» последовательностей предпочтений на этой улице. Разобьем их на группы по $(n+1)$ последовательностей, как в задаче 4.1, получится $(n+1)^{m-1}$ групп. Если в результате парковки $(n+1)$-е место оказалось свободно, то никто из водителей не любит $(n+1)$-е место и расширенная последовательность на самом деле является парковочной последовательностью для исходной улицы. Очевидно, каждая группа содержит в точности $n+1-m$ последовательностей, для которых $(n+1)$-е место оказывается свободным. Поэтому общее число парковочных последовательностей равно $(n+1-m)(n+1)^{m-1}$.

Решение 2 из книги [2, задача 96.54].

Для каждого целого m, $0 \leqslant m \leqslant n$, обозначим через $N(n,m)$ количество последовательностей предпочтения для m водителей, которые приводят к успешной парковке (в частности, $N(n,0)=1$). Докажем индукцией по n, что $N(n,m)=(n+1-m)(n+1)^{m-1}$ при всех m. При $n=1$ утверждение тривиально.

Переход от $n-1$ к n. Пусть k водителей любят места парковки, отличные от первого, а $m-k$ любят первое место ($0 \leqslant k \leqslant m$). Эти k водителей могут быть выбраны C_m^k способами. Нетрудно убедиться, что при заданных предпочтениях успех или неуспех парковки водителей не зависит от того, в каком порядке они приезжают. Поэтому можно считать, что $m-k$ водителей, которые любят первое место парковки, приезжают самыми последними. Эти $m-k$ водителей заведомо сумеют припарковаться — они займут первые $m-k$ свободных мест, оставшихся после первых k водителей. Значит, успех или неуспех парковки определяется только предпочтениями первых k водителей среди отведенных им $n-1$

мест. Значит, при заданном выборе этих k водителей имеется ровно $N(n-1,k)$ удачных последовательностей предпочтений. Таким образом,

$$N(n,k) = \sum_{k=0}^{m} C_m^k \cdot N(n-1,k)\,. \tag{1}$$

Подставим индукционное предположение и преобразуем:

$$N(n,k) = \sum_{k=0}^{m} C_m^k (n-k) n^{k-1} = \sum_{k=0}^{m} C_m^k n^k - \sum_{k=1}^{m} k \cdot C_m^k n^{k-1} = \sum_{k=0}^{m} C_m^k n^k - \sum_{k=1}^{m} m \cdot C_{m-1}^{k-1} n^{k-1} =$$
$$= \sum_{k=0}^{m} C_m^k n^k - m \cdot \sum_{k=0}^{m} C_{m-1}^k n^k = (n+1)^m - m \cdot (n+1)^{m-1} = (n+1-m) \cdot (n+1)^{m-1}\,,$$

что и требовалось. Здесь использовались тривиальная формула $k \cdot C_m^k = m \cdot C_{m-1}^{k-1}$ и бином Ньютона для разложения $(n+1)^m$ и $(n+1)^{m-1}$ по степеням n.

4.5. Пусть $b_1, b_2, \ldots, b_n$ — последовательность предпочтений, выписанная в порядке неубывания. Тогда $b_1 = b_2 = \ldots = b_k = 1$, $2 \leqslant b_{k+1} \leqslant b_{k+2} \leqslant \ldots$. Заметим, что последовательность предпочтений $b_{k+1}, b_{k+2}, \ldots, b_n$ позволяет $n-k$ водителям припарковаться на улице, где имеется $n-1$ парковочное место (это места со 2-го по n-е). Число таких последовательностей подсчитано в предыдущей задаче и равно $k \cdot n^{n-k-1}$. Количество способов выбрать среди n человек k желающих парковаться на первом месте равно C_n^k. Таким образом, общее число паковочных функций, для которых k человек хотели бы парковаться на первом месте, равно $C_n^k \cdot k \cdot n^{n-k-1} = C_{n-1}^{k-1} n^{n-k}$.

«Для проверки» просуммируем найденные выражения по k:

$$\sum_{k=1}^{n} C_{n-1}^{k-1} n^{n-k} = (n+1)^{n-1}$$

(формула бинома). Получилось число парковочных функций. Вот и хорошо.

Как видим, число парковочных функций в этой задаче такое же, как количество лесов на множествое $[n]$, состоящих из k корневых деревьев (задача 3.4).

Приведем биекцию из [13], дающую комбинаторное доказательство этого факта. Пусть дано дерево на $n+1$ вершине с корнем 0. Построим по нему очередь поиска в ширину. Сначала добавим в очередь корень дерева. Далее на каждом шаге вынимаем первую в очереди вершину и добавляем в конец очереди всех ее сыновей в порядке возрастания их номеров. И так далее, пока очередь не опустеет. Пусть A_i — это множество вершин, добавленных в очередь на i-том шаге (это все сыновья некоторой вершины), a_i — число элементов A_i. Если на шаге i ничего не добавили, то множество A_i пусто. Поскольку дерево является связным графом на $n+1$ вершине, то очередь поиска в глубину для него содержит хотя бы один элемент, пока не будет вынута $n+1$ вершина, и опустеет, когда вершины закончатся, то есть a_i удовлетворяют условиям:

$$\begin{cases} a_1 \geqslant 1 \\ a_1 + a_2 - 1 \geqslant 1 \\ \ldots \\ a_1 + a_2 + \ldots + a_k - k \geqslant 1 \\ a_1 + a_2 + \ldots + a_{n+1} = n \end{cases}$$

Определим парковочную функцию по следующему правилу: элементы множества A_i будут номерами машин, которые хотят припарковаться на месте i. Можно доказать, что построенное отображение является биекцией между парковочными функциями для n машин и деревьями на $n+1$ вершине.

4.6. Достаточно описать обратное отображение, т. е. правило, позволяющее по произвольной последовательности $c = (c_1, c_2, \ldots, c_{n-1})$, где все c_i суть остатки по модулю $n+1$, построить парковочную функцию $a = (a_1, \ldots, a_n)$, для которой c является разностной последовательностью. Это легко получается с помощью решения задачи 4.1.

Действительно, если фиксированы все разности $c_i = a_{i+1} - a_i \pmod{n+1}$, существует ровно $n+1$ последовательностей предпочтений для круговой стоянки с такими разностями. В решении 4.1 именно эти $(n+1)$ последовательностей объединены в одну группу и доказано, что в этой группе содержится ровно одна парковочная функция. Очевидно, она-то нам и нужна.

4.7. Приведенная сумма подсчитывает всевозможные парковочные функции. Действительно, чтобы задать парковочную функцию, сначала для каждого парковочного места i выберем число k_i, показывающее, сколько человек любит парковаться на этом месте. Неотрицательные числа k_i должны удовлетворять ограничениям:

$k_n \leqslant 1$, так как при наличии двух желающих парковаться на последнем месте парковка невозможна;

$k_{n-1} + k_n \leqslant 2$, так как при наличии трех желающих парковаться на двух последних местах парковка не состоится;

и т. д. до неравенства $k_2 + \ldots + k_{n-1} + k_n \leqslant n - 1$.

После того как числа k_2, k_3, $\ldots$, k_n выбраны, положим $k_1 = n - k_2 - k_3 - \ldots - k_n$ и назначим последних k_1 человек желающими парковаться на первом месте.

Нетрудно понять, что построенный набор чисел реализуется как парковочная функция. Действительно, по задаче 4.2 порядок, в котором прибывают автомобили, нам неважен. Запуская на парковку, сначала тех, кто хочет парковаться на n-м месте, потом на $(n-1)$-м и т. д., мы прекрасненько всех припаркуем.

Осталось понять, что количество способов реализовать выбранные количества персонально равно стоящему в сумме полиномиальному коэффициенту.

4.8. Решение 1 (из книги [8, задача 5.49.f]). Пусть r_i — число элементов в последовательности a, равных i, в частности, $r_n = 0$. Определение надежной парковочной функции эквивалентно условиям:

1) все частичные суммы последовательности $r_1 - 1$, $r_2 - 1$, $\ldots$, $r_{n-1} - 1$ положительны;

2) $\sum\limits_{i=1}^{n-1} (r_i - 1) = 1$.

Л е м м а. Для любой конечной целочисленной последовательности с суммой 1 существует единственная циклическая перестановка, у которой все частичные суммы положительны.

Очевидно, в надежной последовательности предпочтений все a_i — это числа от 1 до $n-1$. Если мы возьмем произвольный набор чисел из $[n-1]^n$, то определенные этим набором чи́сла r_i удовлетворяют условию 2), но, вообще говоря, не удовлетворяют условию 1). Однако по лемме у этого набора чисел существует, и при том только одна, циклическая перестановка, удовлетворяющая первому условию!

Таким образом, количество надежных парковочных функций в $n-1$ раз меньше числа элементов в $[n-1]^n$.

Решение 2 (прямолинейный подсчет). Рассмотрим произвольную надежную парковочную функцию, пусть для нее k водителей хотят припарковаться на 1 месте ($2 \leqslant k \leqslant n$). Дадим одному из них красную кепку и удалим. Предпочтения оставшихся образуют обычную парковочную функцию на дороге с $n-1$ парковочным местом и $k-1$ желающим припарковаться на первом месте. Таким образом, чтобы получить парковочную функцию, где k водителей хотят припарковаться на 1 месте, нужно выбрать водителя в кепке (n способов) назначить предпочтения оставшимся (по задаче 4.5 это можно сделать $(n-1)^{n-k} C_{n-2}^{k-2}$ способами). Правда, получится не просто парковочная функция, а парковочная функция, наделяющая одого из k претендентов на первое место кепкой. Значит, количество искомых парковочных функций выражается суммой $\sum\limits_{k=2}^{n} \frac{n}{k} \cdot (n-1)^{n-k} C_{n-2}^{k-2}$. Нам остается проверить

тождество
$$\sum_{k=2}^{n}\frac{n}{k}\cdot(n-1)^{n-k}C_{n-2}^{k-2}=(n-1)^{n-1}.$$

Преобразуем левую часть: заменим индекс суммирования k на $i=n-k$, воспользуемся тем, что $\frac{n}{n-i}\cdot C_{n-2}^{i}(n-1)=(n-1-i)C_n^i$ после чего разобьем каждое слагаемое на две части:

$$\sum_{k=2}^{n}\frac{n}{k}\cdot(n-1)^{n-k}C_{n-2}^{k-2}=\sum_{i=0}^{n-2}\frac{n}{n-i}\cdot C_{n-2}^{i}(n-1)^{i}=\sum_{i=0}^{n-2}(n-1-i)\cdot C_n^i(n-1)^{i-1}=$$
$$=\sum_{i=0}^{n-2}C_n^i(n-1)^i-\sum_{i=0}^{n-2}C_n^i\cdot i(n-1)^{i-1}.$$

В последнем выражении первая сумма по формуле бинома равна
$$n^n-C_n^n(n-1)^n-C_n^{n-1}(n-1)^{n-1}. \qquad (2)$$

Для второй суммы отбросим нулевое слагаемое, сделаем преобразование $C_n^i\cdot i=nC_{n-1}^{i-1}$, и тогда она тоже считается с помощью бинома:

$$\sum_{i=0}^{n-2}C_n^i\cdot i(n-1)^{i-1}=n\sum_{i=1}^{n-2}C_{n-1}^{i-1}(n-1)^{i-1}=n\big(n^{n-1}-C_{n-1}^{n-1}(n-1)^{n-1}-C_{n-1}^{n-2}(n-1)^{n-2}\big). \qquad (3)$$

Осталось заметить, что разность выражений (2) и (3) равна $(n-1)^{n-1}$.

4.9. Пусть b_1, b_2, …, b_{n+1} — последовательность номеров любимых мест, выписанная в порядке возрастания. Тот факт, что все сумеют припарковаться, означает выполнение неравенств $b_1\leqslant 1$ (следовательно, $b_1=1$), $b_2\leqslant 2$, …, $b_{n+1}\leqslant n+1$.

Выберем наибольшее k, $0\leqslant k\leqslant n$, для которого $b_{k+1}=k+1$. Тогда последовательность предпочтений b_1, …, b_k является парковочной функцией для улицы с k парковками, а последовательность $b_{k+1}-k$, $b_{k+2}-k$, …, $b_{n+1}-k$ — надежная парковочная функция для улицы с $n+1-k$ парковками. (Надежность следует из максимальности k.) Верно и обратное: выбор парковочной последовательности b_1, …, b_k и надежной парковочной последовательности $b_{k+1}-k$, $b_{k+2}-k$, …, $b_{n+1}-k$ позволяет задать парковочную функцию для все улицы.

Таким образом, чтобы построить произвольную парковочную функцию на улице с $n+1$ парковкой, нужно выбрать любое k, выбрать C_{n+1}^k произвольных k водителей, назначить для их предпочтений одну из P_k парковочных функций, а для остальных $n+1-k$ водителей в качестве препочтений назначить одну из $(n-k)^{n-k}$ надежных парковочных функций. Это и есть комбинаторное истолкование рассматриваемой формулы.

4.10. Аналогично задаче 4.7 формула подсчитывает количество надежных парковочных функций на парковке длины n.

4.11. В [10] эта формула доказана с помощью непростой биекции с помеченными деревьями. Мы докажем ее, пользуясь сходством с предыдущими задачами.

Назовем модернизированной парковочной функцией любую парковочную последовательность $(a_1,\dots,a_n)$, в которой дополнительно пронумерованы все элементы, равные 1. Например, $(\underset{1}{1},2,\underset{2}{1},3,3)$ и $(\underset{2}{1},2,\underset{1}{1},3,3)$ — две различные модернизированные парковочные функции (для $n=5$). Количество модернизированных парковочных функций на парковке длины n обозначим $N^*(n)$.

Рассуждая как в задаче 4.7, мы получаем, что сумма подсчитывает количество модернизированных парковочных функций. Осталось проверить, что $N^*(n)=n^n$.

Рассмотрим модернизированные парковочные последовательности, в которых k человек любят НЕ первое место, а $n-k$ предпочитают парковаться на первом месте. Пользуясь обозначениями и рассуждениями задачи 4.4, заключаем, что число таких последовательностей равно $\frac{n!}{k!}N(n-1,k)$. И тогда суммарное количество модернизированных парковочных функций задается формулой, аналогичной формуле (1):

$$N^*(n) = \sum_{k=0}^{n} \frac{n!}{k!} \cdot N(n-1,k) = \sum_{k=0}^{n} \frac{n!}{k!} \cdot (n-k)n^{k-1}.$$

Это телескопическая сумма, она равна n^n.

5 Инверсии на деревьях и неудобства парковочных функций

5.1. Оба утверждения навеяны [16].

а) Возьмем произвольное помеченное дерево на n вершинах. Пусть n–u — это первое ребро на пути из вершины n в вершину 1 (возможно, $u = 1$). Удалим это ребро, дерево распадется на два дерева A_n и A_1. При этом в дереве A_1 естественным образом выделена вершина u, т.е. можно считать, что это дерево корневое. Пусть дерево A_1 имеет $k+1$ вершину, $0 \leqslant k \leqslant n-2$. Тогда дерево A_n имеет $n-k-1$ вершин. Очевидно, помеченный лес (A_n, A_1) однозначно задается исходным помеченным деревом на n вершинах.

Покажем, как устроено обратное отображение. Для каждого k, $0 \leqslant k \leqslant n-2$, выберем вершины для дерева A_1. Поскольку вершина 1 должна находиться в дереве A_1, а вершина n — в дереве A_n, для этого нужно выбрать k вершин среди всех элементов множества $[n]$, кроме 1 и n; это можно сделать C_{n-2}^k способами. Остальные $n-k-1$ вершин будут принадлежать дереву A_n. Далее построим сами деревья A_1 и A_n — это можно сделать соответственно $(k+1)T_{k+1}$ и T_{n-k-1} способами. Наконец, нам нужно «скрепить» деревья, для этого нужно провести ребро, соединяющее вершину n с любой корнем дерева A_1. Мы получили комбинаторное истолкование k-го слагаемого из суммы в правой части доказываемого тождества: оно равно количеству остовных лесов на множестве $[n]$, состоящих ровно из двух деревьев, таких что одно дерево содержит вершину n и кроме нее еще k других вершин, а другое дерево — корневое и содержит вершину 1.

b) Рассмотрим произвольную парковочную функцию $a = (a_1, a_2, \ldots, a_{n+1})$ для улицы с $n+1$ парковочным местом. Пусть водители въехали на улицу в порядке возрастания номеров и последнему из них досталось место k. Зафиксируем предпочтения первых n водителей и попытаемся увеличить предпочтение последнего водителя, насколько это возможно, т.е. подберем максимальное a_{n+1}, для которого последовательность $(a_1, a_2, \ldots, a_{n+1})$ остается парковочной функцией. Очевидно, этим максимальным значением является $a_{n+1} = k$, причем никто из остальных водителей не претендовал на k-е место, а также никто из тех, кто не смог припарковаться на любимом месте, не проезжал мимо k-го места. Отсюда следует, что ровно $k-1$ человек предпочитали парковаться на местах с 1-го $(k-1)$-е и ровно $n+1-k$ человек предпочитали парковаться на местах с $(k+1)$–го по $(n+1)$-е. Положим, $\widetilde{a} = (a_1, a_2, \ldots, a_n, k)$ и будем называть эту последовательность укрупнением последовательности a.

Ясно, что при любом $\ell \leqslant k$ последовательность $(a_1, a_2, \ldots, a_n, \ell)$ является парковочной функцией, и из каждой из этих k парковочных функций (и только из них) при укрупнении получается фунция $\widetilde{a}$.

Как же нам задать последовательность $\widetilde{a}$? Последний водитель выделен изначально. Нужно выбрать $k-1$ водителя, которые будут парковаться на первых $k-1$ местах (C_n^{k-1} вариантов) и назначить им парковочную функцию (P_{k-1} вариантов). Далее нужно задать парковочную функцию для тех, кто паркуется на местах после k-го ($P_{n-(k-1)}$ вариантов). Итак, для выбора последовательности $\widetilde{a}$ имеется $C_n^{k-1} P_{k-1} P_{n-(k-1)}$ вариантов. Вспоминая, что последовательность $\widetilde{a}$ является укрупнением k различных парковочных функций, заключаем, что число парковочных функций P_{n+1} вычисляется по формуле

$$P_{n+1} = \sum_{k=1}^{n+1} C_n^{k-1} k P_{k-1} P_{n-(k-1)}$$

5.2. Ответ: $F_n^*(x) = \sum\limits_{k=0}^{n} C_n^k F_k(x) F_{n-k}(x)$.

5.3. Оба утверждения взяты из [16]. Эти рекурсии уточняют рекурсии из задачи 5.1.

а) Заметим, что в определении величины $\mathrm{inv}(T)$ можно считать, что метки вершин — это любые вещественные числа. Кроме того, зафиксируем правило, что корень не используется при подсчете числа инверсий дерева, и тогда совершенно неважно, какой меткой он помечен.

Дадим полезное определение. Пусть дано помеченное корневое дерево T на множестве $[\ell]$ и фиксировано множество чисел $S = \{s_1, s_2, \ldots, s_\ell\}$. Определим семейство, состоящее из ℓ корневых деревьев T_1, T_2, …, T_ℓ, вершины которых помечены элементами множества S. Дерево T_i получается из дерева T с помощью замены его меток на метки из множества S следующим образом. В корень дерева T вместо имеющейся там метки ставится метка s_i. Остальные элементы множества S упорядочиваются по возрастанию и расставляются в некорневых вершинах дерева T так, «как это сделано в дереве T»: самая маленькая метка ставится в вершину 1, следующая по величине метка ставится в вершину 2 и т. д. Очевидно, количество инверсий у всех деревьев этого семейства одинаково. Будем называть множество деревьев из этого семейства *равноинверсными* друг другу.

Вернемся к решению задачи. Дополним рассуждения решения 5.1 a) подсчетом инверсий. Чтобы не менять обозначений, докажем требуемое соотношение, взяв вместо параметра n значение $n-2$ (напомним, что многочлен F_n строится по множеству деревьев с $n+1$ вершинами):

$$F_{n-1}(x) = \sum_{k=0}^{n-2} C_{n-2}^k (x^k + x^{k-1} + \ldots + 1) F_k(x) F_{n-k-2}(x).$$

Итак, в решении 5.1 a) всевозможные помеченные деревья находятся во взаимно однозначном соответствии с лесами (A_n, A_1). Чтобы задать лес (A_n, A_1), нужно выбрать k вершин для дерева A_1 среди всех элементов множества $[n]$, кроме 1 и n, (получается C_{n-2}^k вариантов, остальные $n-k-1$ вершин автоматически попадут в дерево A_n) и построить сами деревья A_1 и A_n, при этом дерево A_1 корневое. Далее мы «скрепляем» деревья, проводя ребро, соединяющее вершину n с корнем дерева A_1.

Чтобы подсчитывать инверсии, зафиксируем деревья A_n и A_1 и сгруппируем в один кластер $k+1$ лесов вида (A_n, A), где A пробегает множество корневых деревьев, равноинверсных дереву A_1. Число инверсий в образовавшемся дереве $T \in \mathcal{T}_n$ равно $\mathrm{inv}(A_n) + \mathrm{inv}(A) + \delta_A$, где поправка δ_A равна числу инверсий, которые образует в дереве T корневая вершина дерева A с другими вершинами дерева A (будучи корнем дерева A, она не участвовала в подсчете величины $\mathrm{inv}(A)$, а после скрепления деревьев участвует). Как мы отмечали, $\mathrm{inv}(A) = \mathrm{inv}(A_1)$ для всех деревьев A, равноинверсных дереву A_1. Что же касается поправки δ_A, нетрудно понять, что она по одному разу принимает значения 0, 1, 2, …, k, когда A пробегает множество всех деревьев, равноинверсных дереву A_1.

Каждый лес (A_n, A) из нашего кластера определяет слагаемые $x^{\mathrm{inv}(A_n)}$, $x^{\mathrm{inv}(A)}$ в нумераторах $F_{n-k-2}(x)$, $F_k(x)$. Показатель их произведения $x^{\mathrm{inv}(A_n)+\mathrm{inv}(A)}$ равен числу инверсий в образовавшемся графе без учета поправки. Перебирая все деревья, равноинверсные A_1, получаем произведение $(x^k + x^{k-1} + \ldots + 1)x^{\mathrm{inv}(A_n)+\mathrm{inv}(A)}$, которое дает сумму одночленов, с точностью до перестановки равную слагаемым нумератора $F_{n-1}(x)$ для деревьев, порожденным нашим кластером. Суммируя по всем кластерам, получем требуемую формулу.

b) Дополним рассуждения решения 5.1 b) подсчетом неудобств. В этом решении множество парковочных функций разбито на кластеры. В один кластер с парковочной функцией $a = (a_1, a_2, \ldots, a_{n+1})$ входят парковочные функции $(a_1, a_2, \ldots, a_n, i)$, где $1 \leqslant i \leqslant k$, а через k обозначено максимальное возможное предпочтение $(n+1)$-го водителя, при котором последовательность $(a_1, a_2, \ldots, a_n, k)$ все еще остается парковочной функцией.

Для перечисления всевозможных парковочных функций с заданным параметром k мы перебираем всевозможные функции a' для тех, кто паркуется на первых $k-1$ местах, и всевозможные функции a'' для тех, кто паркуется на последних $n-(k-1)$ местах. Позволяя некоторую вольность речи, скажем, что неудобство этих парковочных выриантов равно $D(a')$ и $D(a'')$, а на языке нумераторов — $x^{D(a')}$ и $x^{D(a'')}$. Что же касается $(n+1)$-го человека, который по нашему плану паркуется на k-м месте, то его вклад в общее неудобство принимает для функций из нашего кластера по одному разу значения 0, 1, …, $k-1$. Таким образом, суммарное неудобство парковочных функций из одного кластера записывается выражением $(x^{k-1} + x^{k-2} + \ldots + 1)x^{D(a')}x^{D(a'')}$. Суммируя по всем кластерам, получаем требуемую формулу.

5.4. Требуемая биекция коротко описана в [19]. Следующую конструкцию, которая привносит в эти рассуждения дополнительные подробности, нам любезно сообщил И. Богданов.

Рассмотрим произвольное помеченное дерево T с $n+1$ вершиной без инверсий. Как обычно, мы считаем вершину $n+1$ корнем, вводя тем самым отношение «предок–потомок». Припишем к каждой вершине, кроме корня, целое неотрицательное число, не превосходящее числа её потомков (исключая её саму); это число будем называть *плохостью* вершины. Полученный объект (помеченное дерево с приписанными плохостями) назовём *оснащённым деревом* (или кратко *о-деревом*), а множество всех оснащённых деревьев с $n+1$ вершиной обозначим через $\mathcal{E}_{n+1}$. Сумму всех плохостей вершин о-дерева E назовём *плохостью* о-дерева и обозначим $\mathrm{bad}(E)$.

Требуемая биекция является композицей двух биекций, описанных ниже.

(i): Биекция между $\mathcal{E}_{n+1}$ и $\mathcal{T}_{n+1}$. **Только** в рамках построения этой биекции мы будем **различать** *вершину* дерева и *число*, которым она помечена (т. е. *пометку* вершины). Это полезно, ибо в процессе построения биекции пометки будут переставляться.

Назовём *инверсностью* вершины помеченного дерева количество её потомков, числа в которых больше, чем число в самой вершине. Ясно, что количество инверсий в дереве равно сумме инверсностей его вершин.

Мы устроим биекцию следующим образом. По о-дереву $E \in \mathcal{E}_{n+1}$ мы построим дерево $T \in \mathcal{T}_{n+1}$ такое, что

(a) T и E соответствуют одному и тому же *непомеченному* дереву, и их корни совпадают;

(b) плохость любой *вершины* в E равна её инверсности в T.

Как следствие, $\mathrm{bad}(E) = \mathrm{inv}(T)$.

Для этого в о-дереве E надо переставить пометки вершин. Мы будем делать это «сверху вниз», обработав сначала всех сыновей корня, затем всех их сыновей и т. д. Пусть v — очередная вершина, которую надо обработать, i — её плохость в о-дереве, а $a_1 > a_2 > \ldots > a_k$ — текущие пометки её и всех её потомков в убывающем порядке (тогда a_1 — пометка в v, и $k > i$). Переставим метки $a_1, \ldots, a_{i+1}$ так: поставим a_{i+1} в v, а a_{j+1} заменим на a_j при всех $j = 1, 2, \ldots, i$. Нетрудно понять, что инверсность v стала равна i, никакие два потомка v инверсии не образуют, и инверсности остальных вершин не поменялись (ибо не поменялись множества их потомков). Значит, обработав таким образом все вершины, мы получим дерево T с требуемыми свойствами.

На рисунке ниже показан пример действия этого алгоритма (индексы указывают плохости вершин).

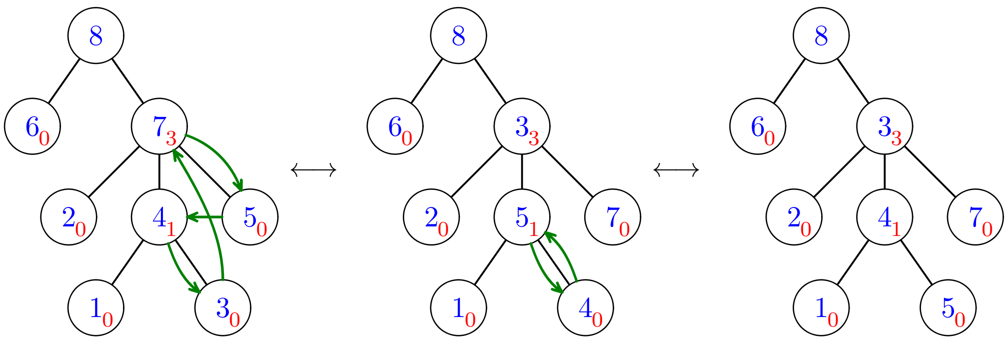


Рис. 7. Построение T по E (и наоборот)

Чтобы построить обратное отображение из $\mathcal{T}_{n+1}$ в $\mathcal{E}_{n+1}$, заметим, что плохости вершин в E расставляются автоматически, исходя из условия (b). Перестановка же пометок строится похожим образом снизу вверх. Когда обрабатывается вершина v, никакие два её потомка не образуют инверсии. Пусть $a_1 > \ldots > a_k$ — пометки её и её потомков, причём в v стоит a_{i+1}. Тогда достаточно переставить метки по циклу $a_1 \to a_{i+1} \to a_i \to \ldots \to a_1$. В результате этого процесса получится требуемое о-дерево.

Легко видеть, что построенные отображения взаимно обратны.

(ii): Биекция между $\mathcal{P}_n$ и $\mathcal{E}_{n+1}$. Пусть $a = (a_1, \dots, a_n)$ — парковочная функция, а $p_1, \dots, p_n$ — номера мест, на которых припарковались соответствующие водители; тогда $p = (p_1, \dots, p_n)$ — перестановка чисел $1, 2, \dots, n$. Назовём *неудобством* i-го водителя величину $p_i - a_i$; суммарное неудобство всех водителей есть неудобство функции a.

Мы устроим биекцию следующим образом. По функции a мы построим о-дерево $E \in \mathcal{E}_{n+1}$ такое, что

(c) плохость вершины с пометкой k равна неудобству водителя k.

Как следствие, $\mathrm{bad}(E) = D(a)$, а значит, композиция двух наших биекций — требуемая.

Осталось, собственно, предъявить вторую биекцию. Мы начнём с более простой биекции — между перестановками $p = (p_1, \dots, p_n)$ и помеченными деревьями $T \in \mathcal{T}_{n+1}$ без инверсий. Затем мы продолжим эту биекцию на парковочные функции и оснащённые деревья.

Шаг 1. Сопоставление перестановок деревьям без инверсий. Положим $p_{n+1} = n + 1$ и рассмотрим числа $1, 2, \dots, n+1$ как вершины строящегося дерева; назовём число p_i *местом* вершины i. Для каждой вершины $i \leqslant n$ найдём наименьшее из мест p_j при $j > i$, которое больше p_i, и соединим i с j. Мы получим связный граф (все вершины соединены путём с $n + 1$) с n рёбрами, т. е. дерево T. (Процесс получения этого дерева удобно представлять себе «справа налево»: мы по очереди рассматриваем вершины $n, n-1, \dots, 1$ и подсоединяем их к получающемуся справа дереву согласно правилу.)

Исследуем полученное дерево. Ясно, что в описанной выше ситуации i — сын j. Выберем произвольную вершину $j \leqslant n$. Для любого потомка k вершины j выполнены неравенства $k < j$ и $p_k < p_j$ (в частности, в дереве нет инверсий). Пусть $s > j$ — вершина, для которой $p_s < p_j$, и p_s максимальное возможное (если такого нет, положим $p_s = 0$). Тогда, рассмотрев процесс получения дерева справа налево, нетрудно видеть, что выполнено следующее свойство:

(d) места всех потомков вершины j больше p_s и меньше p_j,

и наоборот — все вершины с такими местами являются потомками j (ибо они присоединяются либо к j, либо к её потомкам).

Свойства (c) и (d) позволяют по дереву T восстановить перестановку p, действуя опять же справа налево. Начнём с вершины n; согласно (d), места её и её потомков в точности составляют число от 1 до p_n; поэтому p_n есть число её потомков. Далее действуем аналогично: рассматривая очередную вершину j, являющуюся сыном некоторой вершины k, мы знаем, что p_j должно лежать между p_s и p_k, где p_s — наибольшее уже определённое место, меньшее p_k. Значит, если у j ровно d потомков, то они будут иметь места $p_s + 1, \dots, p_s + d$, а само p_j равно $p_s + d + 1$. Таким образом, перестановка (p_i) восстановлена.

На рисунке ниже показана работа обоих алгоритмов. Около вершин дерева справа указаны восстанавливаемые значения p_i, а также определяемый диапазон мест их потомков.

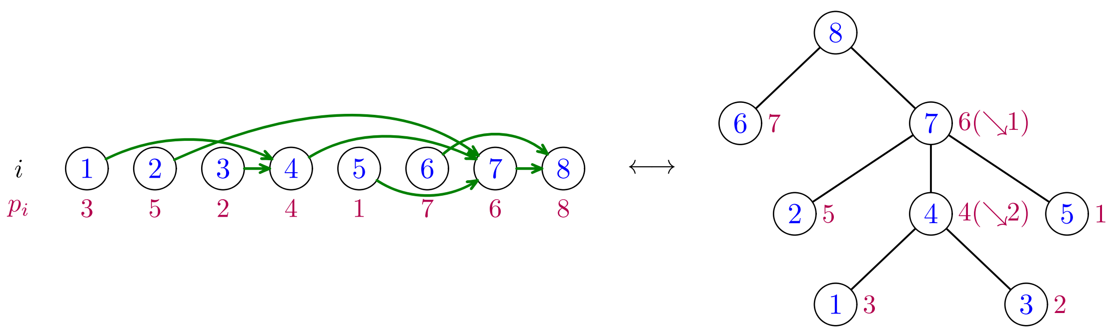


Рис. 8. Построение T по p (и наоборот)

Теперь нетрудно видеть, что построенные отображения действительно взаимно обратны. Действительно, отображение $T \mapsto p$ строилось так, чтобы восстановить исходную перестановку, если T было по ней построено. Наоборот, пусть мы построили p по T. Выберем любое $i \leqslant n$, являющееся сыном k в дереве T. Пусть $j > i$ — та вершина, к которой i подсоединится в обратной процедуре. Тогда $p_i < p_j \leqslant p_k$, то есть j — потомок k или сама k. С другой стороны, по построению p, все потомки k, лежащие между i и k, получали места, меньшие p_i. Значит, $j = k$, и по перестановке p восстанавливается именно T.

Шаг 2: Оснащение. Теперь можно построить требуемую биекцию. Пусть a — парковочная функция, а p — соответствующая ей перестановка. Тогда по p можно построить помеченное дерево T и оснастить его согласно (c), сопоставив вершине j плохость $p_j - a_j$. Согласно (d), число потомков j равно $p_j - p_s - 1$. С другой стороны, когда машина j попала на место p_j, место p_s было свободным; значит, $a_j > p_s$, так что $p_j - a_j \leqslant p_j - p_s - 1$, т. е. плохость вершины j не превосходит числа её потомков. Итого, мы получили о-дерево E. Наоборот, по о-дереву E (точнее, по подлежащему помеченному дереву T) можно восстановить перестановку p, а по ней — функцию a, опять же согласно (c). Ясно, что если по a псстроить E, то по E восстановится именно функция a. Осталось понять, что, наоборот, если по произвольному $E \in \mathcal{E}_{n+1}$ построить функцию a, то по ней обратно построится дерево E.

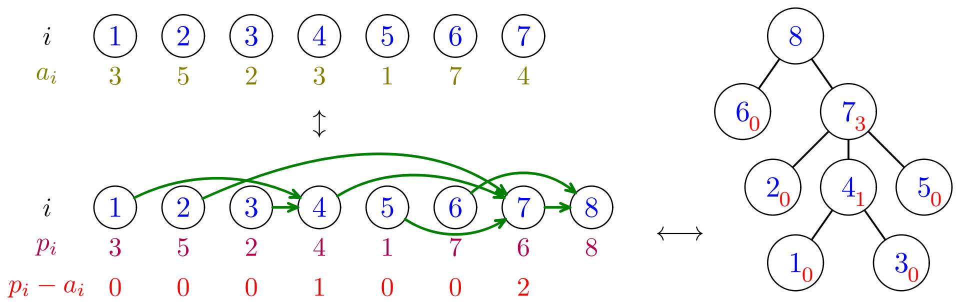


Рис. 9. Оснащение дерева

Для начала покажем, что полученная функция a — парковочная. Поскольку $a_i \leqslant p_i$, а (p_i) — перестановка, для этого достаточно показать, что $a_i > 0$. Это просто: если у вершины i есть d потомков, все они имеют номера, меньшие p_i; значит, p_i больше, чем d — и, как следствие, чем плохость вершины j. Отсюда и следует требуемое.

Осталось показать, что, если по о-дереву E получены перестановка $p = (p_i)$ и парковочная функция a, то, действуя по этой парковочной функции, водители образуют ровно перестановку p. Для начала вспомним, что согласно (d) места p_j всех потомков j вершины i и её самой образуют множество $A_j = \{p_i - d, \ldots, p_j - 1, p_j\}$, где d — число потомков j. Значит, $a_j \in A_j$.

Теперь нетрудно показать, что водитель i попадёт на место p_i, индукцией по i. При $i = 1$ это верно, ибо 1 — лист нашего дерева, то есть $p_1 = a_1$. Пусть все водители $j < i$ попали на предписанные места. В частности, попали на свои места все потомки i; значит, все места $p_i - d, \ldots, p_i - 1$ уже заняты, а p_i — свободно. Так как $a_i \in A_i$, отсюда и следует, что водитель i припаркуется на p_i. Доказательство окончено.

Литература

[1] *Айгнер М., Циглер Г.* Доказательства из Книги. М.:Мир, 2006.

[2] *Берлов С. Л., Иванов С. В., Кохась К. П.* Петербургские математические олимпиады. СПб.: «Лань», 2000.

[3] *Глибичук А. А., Дайняк А. Б., Ильинский Д. Г., Купавский А. Б., Райгородский А. М., Скопенков А. Б., Чернов А. А.* Элементы дискретной математики в задачах. М.:МЦНМО, 2016.

[4] *Иванов О. А.* Элементарная математика для школьников, студентов и преподавателей. М.:МЦНМО, 2009.

[5] Комбинаторный анализ. Задачи и упражнения. / Под. ред. К. А. Рыбникова. М.: Наука, 1982.

[6] *Ландо С. К.* Введение в дискретную математику. М.: МЦНМО, 2012.

[7] *Сачков В. Н.* Введение в комбинаторные методы дискретной математики. М.: МЦНМО, 2004.

[8] *Стэнли Р.* Перечислительная комбинаторика. Т. 2. М.: Мир, 2005.

[9] *Beineke L. W., Moon J. W.* Several proofs of the number of labelled 2-dimensional trees. In «Proof Techniques in Graph Theory» (F. Harary, editor). New York: Academic Press, 1969. P. 11-20.

[10] *Benjamin A. T., Juhnke F.* Another way of counting N^N. // SIAM J. Disc. Math. 1992. V. 5. No. 3. P. 377–379.

[11] *Bóna M.* Combinatorics of permutations. Chapman & Hall/CRC, 2004.

[12] *Bóna M.* A walk through combinatorics. 3rd ed. World Scientific Publ. Co., 2011.

[13] *Chassaing P. , Marckert J.-F.* Parking functions, empirical processes, and the width of rooted labeled trees // Electron. J. Combin. 2001. V. 8. № 1. Research Paper 14.

[14] *Guo S., Guo V.* A recursive algorithm for trees and forests // `http://arxiv.org/pdf/math/1702.01744v1.pdf`

[15] *Konheim A. G., Weiss B.* An occupancy discipline and applications. // SIAM J. Appl. Math. 1966. Vol. 14. № 6. P. 1266–1274.

[16] *Kreweras G.* Une famille de polynômes ayant plusieurs propriétés énumeratives // Period. Math. Hungar. V. 11. 1980. № 4. P. 309–320.

[17] *Moon J. W.* Counting labelled trees. William Clowes and Sons, 1970.

[18] *Shukla A.* A short proof of Cayley's tree formula // `https://arxiv.org/pdf/0908.2324.pdf`

[19] *Yan C. H.* Parking functions// in Handbook of Enumerative Combinatorics. M. Bóna, ed.