

Деревья

Деревья представляют собой важный вид графов. С помощью деревьев описываются базы данных, деревья моделируют алгоритмы и программы, их используют в электротехнике, химии.

Деревом называется связный неориентированный граф без циклов (ациклический), который содержит более двух вершин.

Остовное дерево графа (остов) - дерево, которое содержит все вершины исходного графа.

Число остовов графа можно вычислить по его матричным характеристикам.

Теорема (Кирхгофа). Число остовов графа равно алгебраическому дополнению любого элемента матрицы Кирхгофа.

Элементы матрицы Кирхгофа B графа G определяются следующим образом:

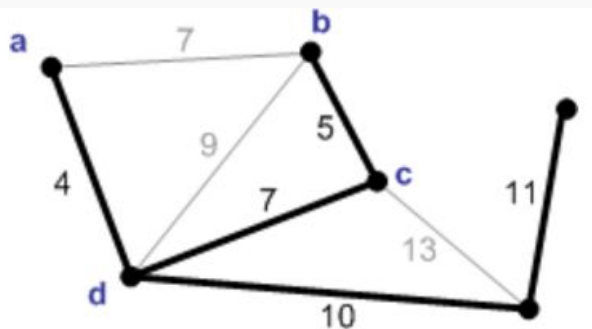
$$b_{ij} = \begin{cases} -1 & \text{если вершины } i \text{ и } j \text{ смежны;} \\ 0 & \text{если вершины } i \text{ и } j \text{ не смежны;} \end{cases} \quad i \neq j,$$
$$b_{ii} = \deg(i)$$

Свойства матрицы Кирхгофа:

- 1) сумма элементов в каждой строке и каждом столбце матрицы Кирхгофа равна 0;
- 2) алгебраические дополнения всех элементов матрицы Кирхгофа равны;
- 3) ранг матрицы Кирхгофа неографа порядка n можно вычислить по формуле: $\text{rang } B = n - k$, где k — число компонент графа.

Минимальное остовное дерево

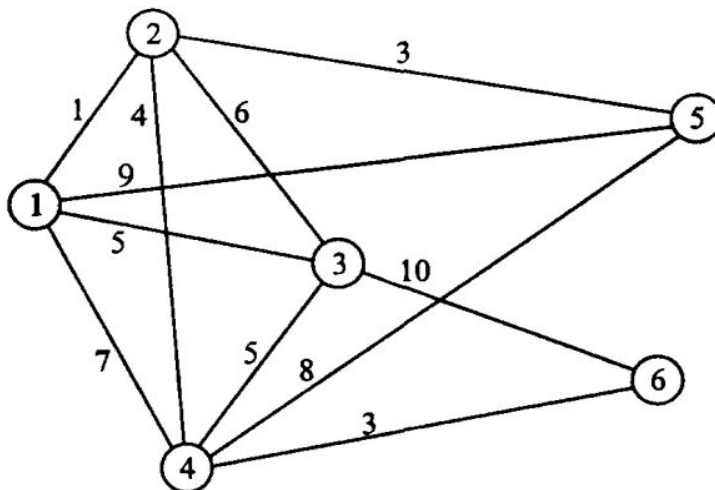
Остовное дерево, у которого суммарный вес его ребер минимален, называется **минимальным остовным деревом**.



На рисунке изображено минимальное остовное дерево, суммарный вес которого равен 37.

Это минимальное остовное дерево не уникально: удалением ребра (c,d) и добавлением ребра (a,b) получается новое минимальное остовное дерево.

Задача. Телевизионная компания планирует подключение к своей кабельной сети пяти новых районов. На рисунке показана структура планируемой сети и расстояния (в милях) между районами и телецентром. Необходимо спланировать наиболее экономичную кабельную сеть.



Алгоритм построения минимального остовного дерева

Опишем процедуру выполнения этого алгоритма. Обозначим через $N = \{1, 2, \dots, n\}$ множество узлов сети и введем новые обозначения:

C_k — множество узлов сети, соединенных алгоритмом после выполнения k -й итерации этого алгоритма,

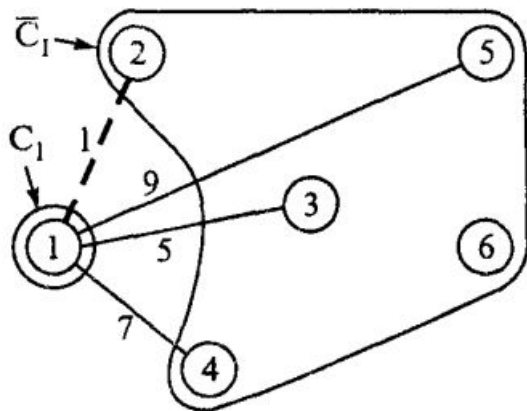
$\bar{C}_k$ — множество узлов сети, не соединенных с узлами множества C_k после выполнения k -й итерации этого алгоритма.

Шаг 0. Полагаем $C_0 = \emptyset$ и $\bar{C}_0 = N$.

Шаг 1. Выбираем *любой* узел i из множества $\bar{C}_0$ и определяем $C_1 = \{i\}$, тогда $\bar{C}_1 = N - \{i\}$. Полагаем $k = 2$.

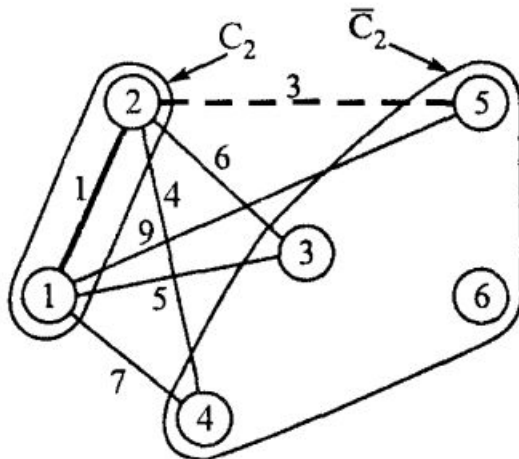
Основной шаг k . В множестве $\bar{C}_{k-1}$ выбираем узел j^* , который соединен самой короткой дугой с каким-либо узлом из множества C_{k-1} . Узел j^* присоединяется к множеству C_{k-1} и удаляется из множества $\bar{C}_{k-1}$. Таким образом, $C_k = C_{k-1} + \{j^*\}$, $\bar{C}_k = \bar{C}_{k-1} - \{j^*\}$.

Если множество $\bar{C}_k$ пусто, то выполнение алгоритма заканчивается. В противном случае полагаем $k = k + 1$ и повторяем последний шаг.



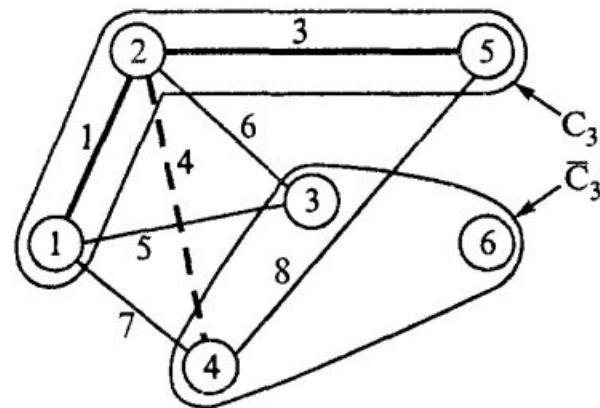
Итерация 1

$$C_1 = \{1\}; \bar{C}_1 = \{2, 3, 4, 5, 6\}$$



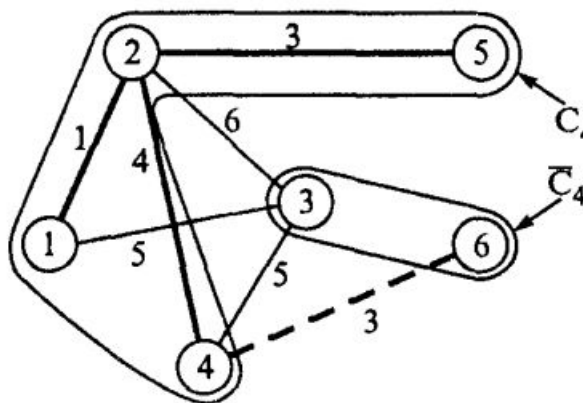
Итерация 2

$$C_2 = \{1, 2\}; \bar{C}_2 = \{3, 4, 5, 6\}$$



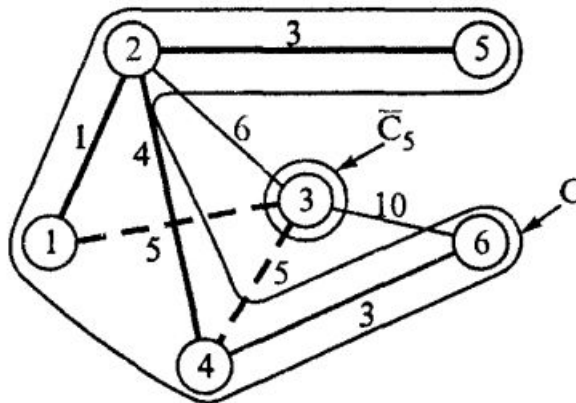
Итерация 3

$$C_3 = \{1, 2, 5\}; \bar{C}_3 = \{3, 4, 6\}$$



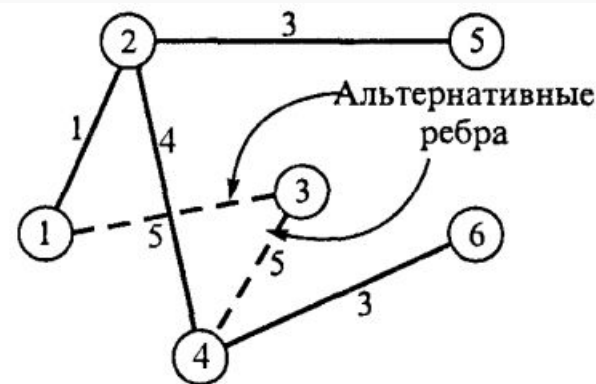
Итерация 4

$$C_4 = \{1, 2, 4, 5\}; \bar{C}_4 = \{3, 6\}$$



Итерация 5

$$C_5 = \{1, 2, 4, 5, 6\}; \bar{C}_5 = \{3\}$$



Итерация 6

(Минимальное остовное дерево)

Решение в виде минимального остовного дерева получено на 6-й итерации.

Минимальная длина кабеля для построения такой сети равна:

$$1+3+4+3+5 = 16 \text{ милям.}$$

Существует несколько алгоритмов для нахождения минимального остовного дерева.

Некоторые наиболее известные из них:

Алгоритм Борувки (1926)

Алгоритм Прима (1956)

Алгоритм Краскала (или алгоритм Крускала) (1956)

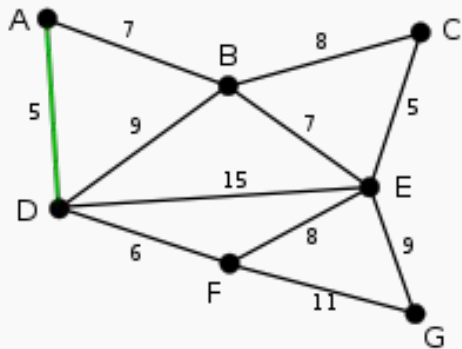
Алгоритм Дж. Краскала – эффективный алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа.

Вначале текущее множество рёбер устанавливается пустым.

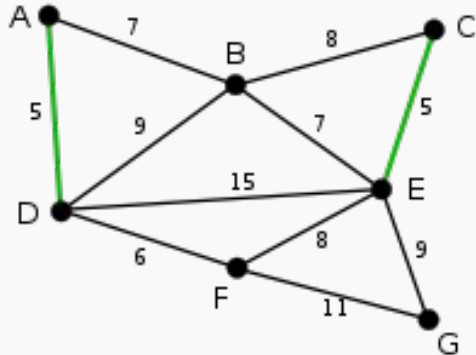
Затем, пока это возможно, проводится следующая операция: из всех рёбер, добавление которых к уже имеющемуся множеству не вызовет появление в нём цикла, выбирается ребро минимального веса и добавляется к уже имеющемуся множеству.

Когда таких рёбер больше нет, алгоритм завершён.

Подграф данного графа, содержащий все его вершины и найденное множество рёбер, является его **остовным деревом минимального веса**.

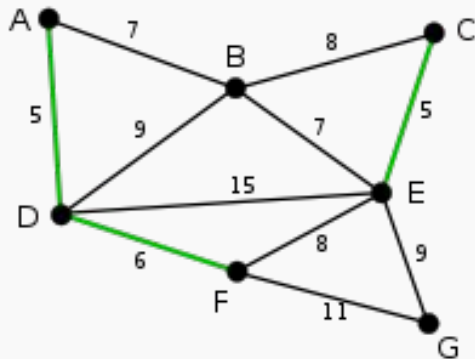


Ребра **AD** и **CE** имеют минимальный вес, равный 5. Произвольно выбирается ребро **AD** (выделено на рисунке).

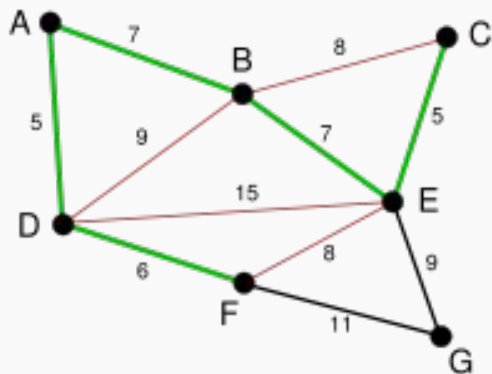


Теперь наименьший вес, равный 5, имеет ребро **CE**.

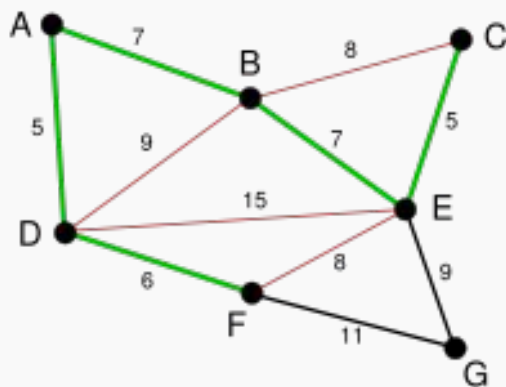
Так как добавление **CE** не образует цикла, то выбираем его в качестве второго ребра.



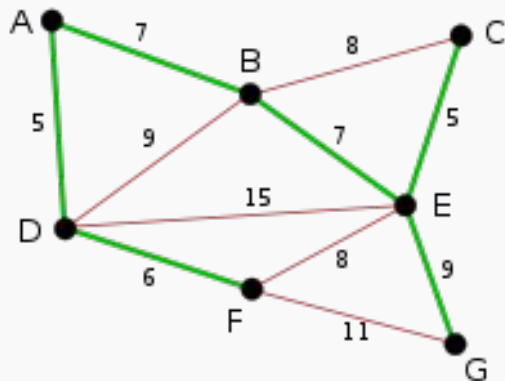
Аналогично выбираем ребро **DF**, вес которого равен 6.



Следующие ребра — **AB** и **BE** с весом 7. Произвольно выбирается ребро **AB**, выделенное на рисунке. Ребро **BD** выделено красным, так как уже существует путь (зелёный) между **A** и **D**, поэтому, если бы это ребро было выбрано, то образовался бы цикл **ABD**.



Аналогичным образом выбирается ребро **BE**, вес которого равен 7. На этом этапе красным выделено гораздо больше ребер: **BC**, потому что оно создаст цикл **BCE**, **DE**, потому что оно создаст цикл **DEBA**, и **FE**, потому что оно сформирует цикл **FEBAD**.



Алгоритм завершается добавлением ребра **EG** с весом 9. Минимальное остовное дерево построено построено.

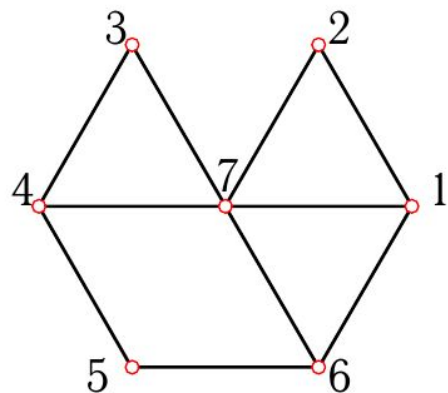
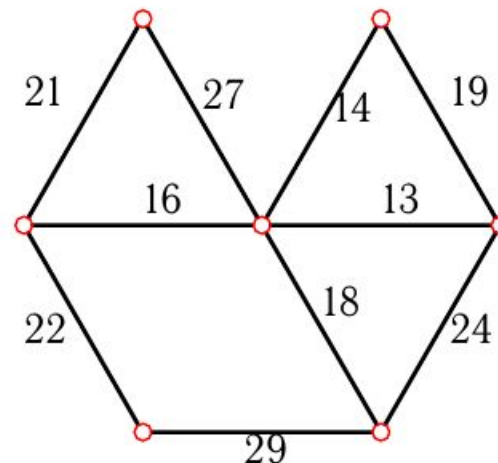
Задача 1. Дан взвешенный граф. Найти

- 1) число остовов графа;
- 2) минимальное остовное дерево.

Решение:

- 1) Пронумеруем вершины графа.

В соответствии с определением запишем матрицу Кирхгофа:



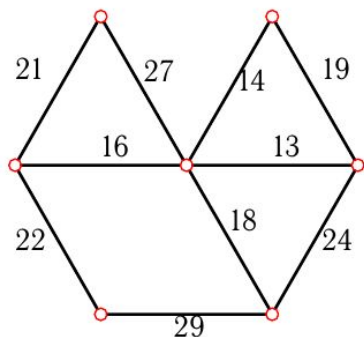
$$B = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & -1 & -1 \\ -1 & 2 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 2 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 & 3 & -1 \\ -1 & -1 & -1 & -1 & 0 & -1 & 5 \end{bmatrix}$$

Рассмотрим минор элемента b_{11} :

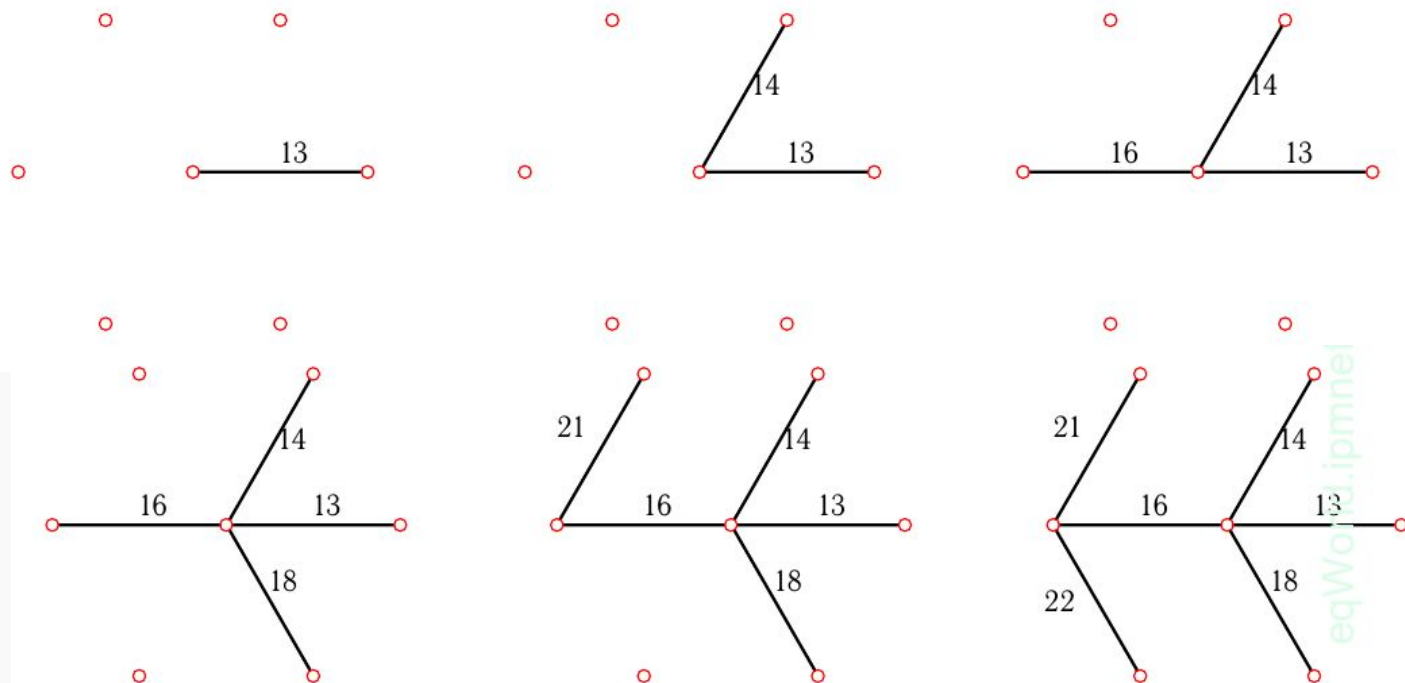
$$M_{11} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & -1 \\ 0 & 2 & -1 & 0 & 0 & -1 \\ 0 & -1 & 3 & -1 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 \\ -1 & -1 & -1 & 0 & -1 & 5 \end{bmatrix}$$

Вычислим определитель: $\det M_{11} = 79$.

Таким образом, граф имеет 79 остовов, среди которых надо выбрать экстремальное дерево, т.е. дерево, обладающее некоторыми экстремальными свойствами, в данном случае — минимальным весом.



2) Строим минимальное остовное дерево при помощи алгоритма Краскала.



Получили минимальное остовное дерево.

Суммарный вес дерева равен: $21+22+16+14+13+18 = 104$.

Центроид дерева

Ветвь к вершине v дерева — это максимальный подграф, содержащий v в качестве висячей вершины. Вес c_k вершины k — наибольший размер ее ветвей.

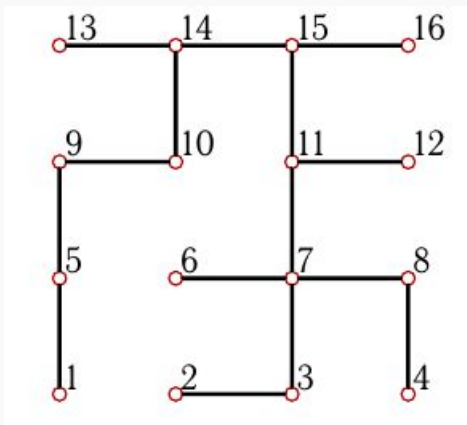
Центроид дерева C (или центр масс дерева) — множество вершин с наименьшим весом: $C = \{v \mid c(v) = c_{\min}\}$

- Вес любого листа дерева равен размеру дерева.
- Высота дерева с корнем, расположенным в центроиде, не больше наименьшего веса его вершин.
- Свободное дерево порядка n с двумя центроидами имеет четное количество вершин, а вес каждого центроида равен $n/2$.

Для центроида дерева существует теорема:

Теорема (Жордана). Каждое дерево имеет центроид, состоящий из одной или двух смежных вершин

Пример. Найти наименьший вес вершин дерева и его центроид.



Решение. Очевидно, вес каждой висячей вершины дерева порядка n равен $n - 1$.

Висячие вершины не могут составить центроид дерева, поэтому исключим из рассмотрения вершины 1, 2, 4, 6, 12, 13 и 16.

Для всех остальных вершин найдем их вес, вычисляя длину (размер) их ветвей.

Число ветвей вершины равно ее степени. Размеры ветвей вершин 3, 5 и 8 равны 1 и 14. Следовательно, веса этих вершин равны 14.

К вершине 7 подходят четыре ветви размером 1, 2, 2 и 10. Таким образом, ее вес $c_7 = 10$.

Аналогично вычисляются веса других вершин:

$$c_9 = 13, \quad c_{10} = c_{14} = 12, \quad c_{11} = c_{15} = 8.$$

Минимальный вес вершин равен 8, следовательно, центроид дерева образуют две вершины с таким весом: 11 и 15.

Кодировка деревьев

Одной из актуальных задач в эпоху компьютерных и телекоммуникационных сетей является задача сжатия информации. Сюда входит и кодировка деревьев.

Компактная запись дерева, полностью описывающая его структуру, может существенно упростить как передачу информации о дереве, так и работу с ним.

Десятичная кодировка

Одна из простейших кодировок помеченных деревьев с выделенным корнем — десятичная.

Кодируя дерево, придерживаемся следующих правил:

1. Кодировка начинается с корня и заканчивается в корне.
2. Каждый шаг на одну дугу от корня кодируется единицей.
3. В узле выбираем направление на вершину с меньшим номером.
4. Достигнув листа, идем назад, кодируя каждый шаг нулем.
5. При движении назад в узле всегда выбираем направление на не пройденную вершину с меньшим номером.

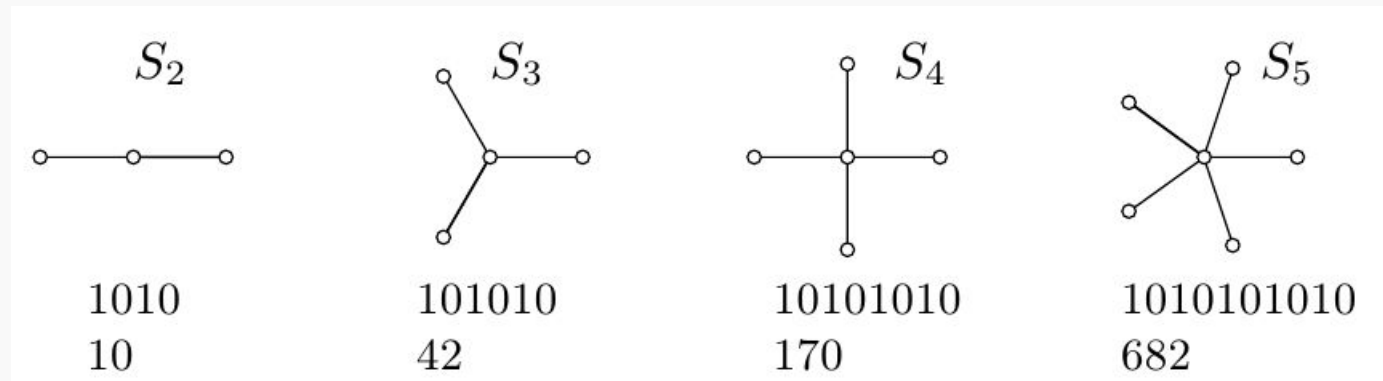
Кодировка в такой форме получается достаточно компактной, однако она не несет в себе информации о номерах вершин дерева.

Существуют аналогичные кодировки, где вместо единиц в таком же порядке проставляются номера или названия вершин.

Есть деревья, для которых несложно вывести формулу десятичной кодировки. Рассмотрим, например, графы-звезды $K_{1,n}$, являющиеся полными двудольными графами, одна из долей которых состоит из одной вершины 1. Другое обозначение звезд — $K_{1,n} = S_n$.

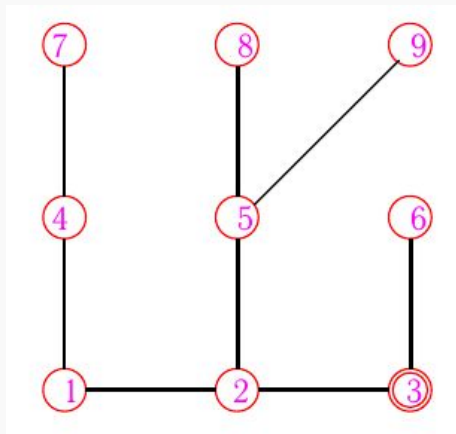
На рисунках приведены звезды и их двоичные и десятичные кодировки. Корень дерева располагается в центральной вершине звезды.

Легко получить общую формулу: $Z(S_n) = 2(4^n - 1)/3$.



Если корень поместить в любой из висячих вершин, то код Z^* такого дерева будет выражаться большим числом. Более того, существует зависимость $Z(S_n) - Z^*(S_n) = Z(S_n - 1)$.

Пример. Записать десятичный код дерева с корнем в вершине 3.



Решение. На основании правила кодировки, двигаясь по дереву, проставим в код единицы и нули. При движении из корня 3 к вершине 7 проходим четыре ребра. В код записываем четыре единицы: 1111.

Возвращаясь от вершины 7 к вершине 2 (до ближайшей развилки), проходим три ребра. Записываем в код три нуля: 000.

От вершины 2 к 5 и далее к 8 (меньший номер): 11; от 8 назад к 5 и от 5 к 9: 01; от 9 к корню 3: 000.

И наконец, от 3 к 6 и обратно: 10.

В итоге, собирая все вместе, получим двоичный код дерева:

1 111 000 110 100 010.

Разбивая число на тройки, переводим полученное двоичное представление в восьмеричное 1. Получаем 17006428.

Затем переводим это число в десятичное:

$$2 \cdot 80 + 4 \cdot 81 + 6 \cdot 82 + 0 \cdot 83 + 7 \cdot 84 + 1 \cdot 85 = 61858.$$

Можно перевести двоичное число из n цифр в десятичное число непосредственно по формуле:

$$\sum_{i=1}^n k_i 2^{n-i}, \quad \text{где } k_i \text{ — } i\text{-я цифра (0 или 1) в двоичном числе}$$

Кодировка Прюфера

Выбор кодировки дерева зависит от решаемой теоретической или технической задачи. Среди всех возможных кодировок естественно отыскать оптимальные по какому-то качеству решения.

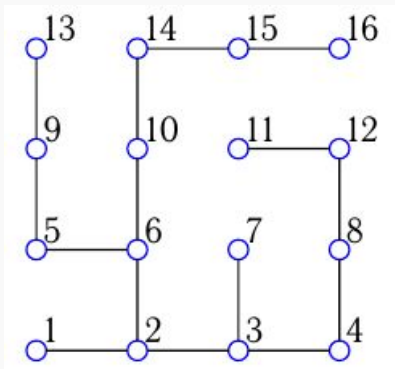
Было показано, что существует оптимальный в определенном смысле код дерева — так называемый код Прюфера

Кодировка Прюфера применяется к свободным деревьям (неориентированным деревьям, т.е. деревьям, в которых нет выделенного корня).

Приведем алгоритм кодировки помеченного дерева по Прюферу:

1. Найти висячую вершину с минимальным номером i .
2. Записать в код Прюфера вершину, смежную с i .
3. Удалить вершину i из дерева. Если дерево не пустое, то перейти к п.1.

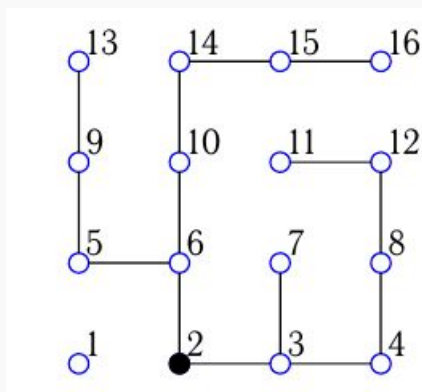
Пример. Записать код Прюфера для дерева



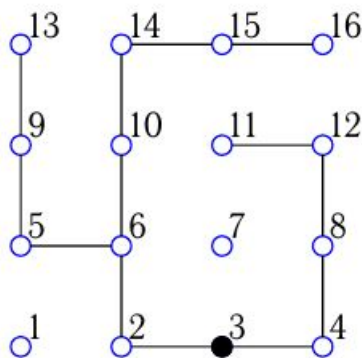
Решение. Находим висячую вершину с минимальным номером, записываем в код Прюфера вершину, смежную с ней, и удаляем ее из дерева.

Последовательность определения кода Прюфера для рассматриваемого дерева показана на рисунках.

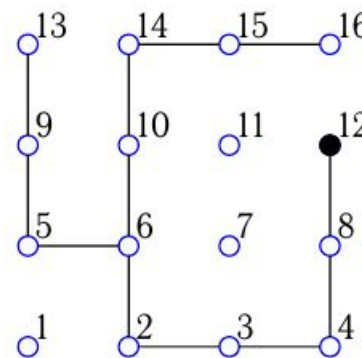
Для наглядности изображение удаленной вершины остается на рисунке, а стирается только ребро.



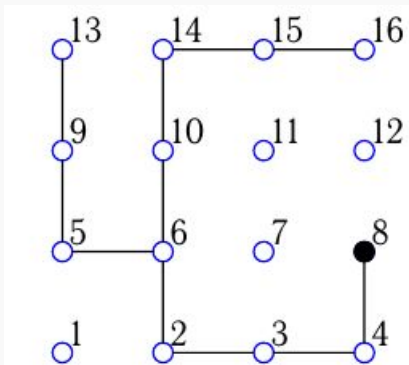
$P = [2]$



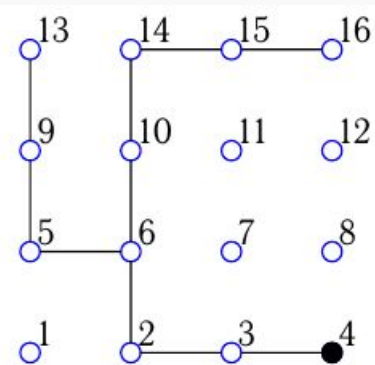
$P = [2, 3]$



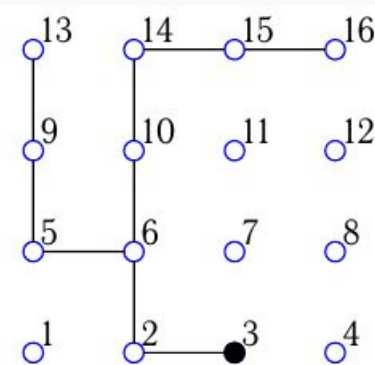
$P = [2, 3, 12]$



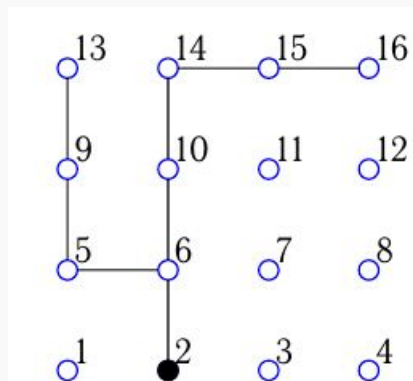
$P = [2, 3, 12, 8]$



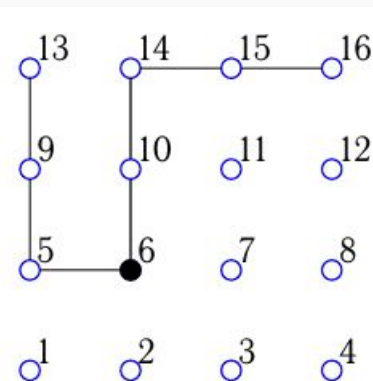
$P = [2, 3, 12, 8, 4]$



$P = [2, 3, 12, 8, 4, 3]$



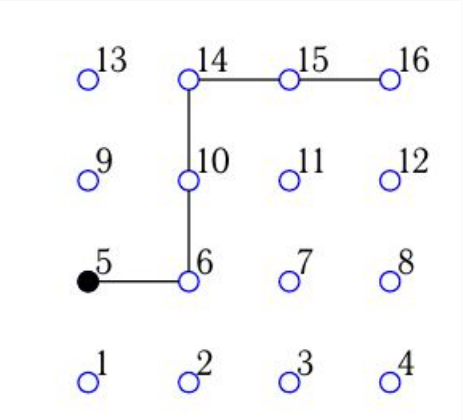
$P = [2, 3, 12, 8, 4, 3, 2]$



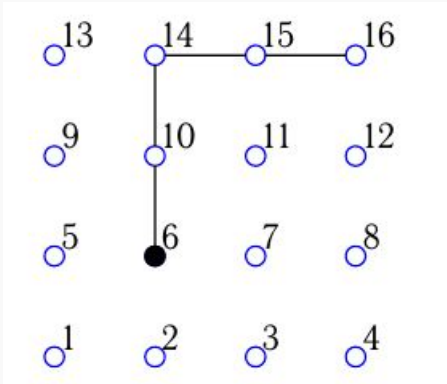
$P = [2, 3, 12, 8, 4, 3, 2, 6]$



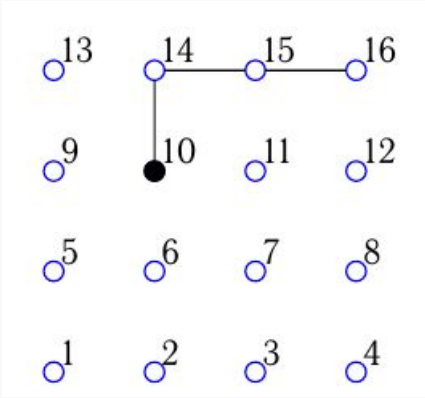
P = [2, 3, 12, 8, 4, 3, 2, 6, 9]



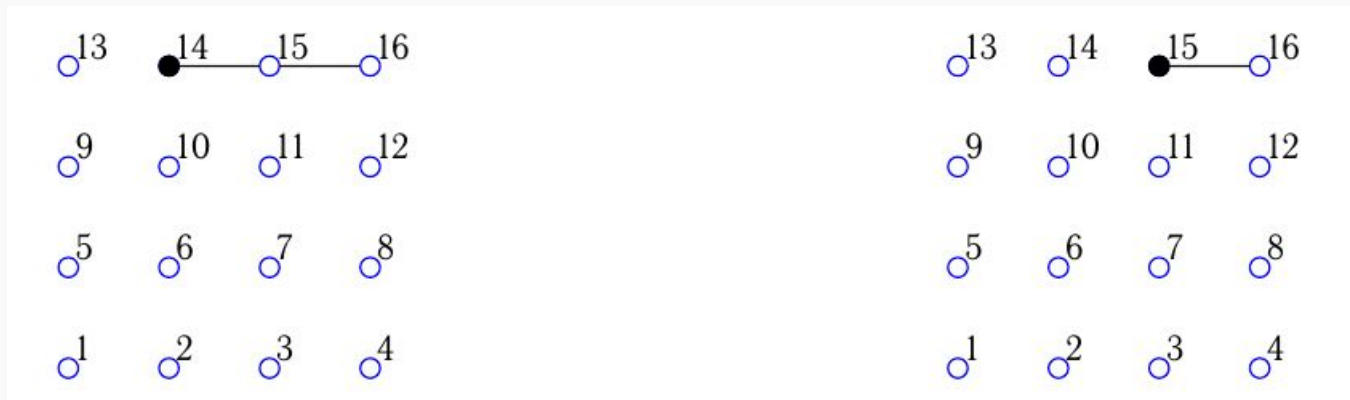
$P = [2, 3, 12, 8, 4, 3, 2, 6, 9, 5]$



$P = [2, 3, 12, 8, 4, 3, 2, 6, 9, 5, 6]$



$P = [2, 3, 12, 8, 4, 3, 2, 6, 9, 5, 6, 10]$



$P = [2, 3, 12, 8, 4, 3, 2, 6, 9, 5, 6, 10, 14]$

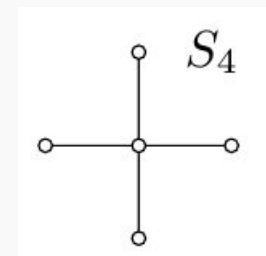
$P = [2, 3, 12, 8, 4, 3, 2, 6, 9, 5, 6, 10, 14, 15]$

В результате код Прюфера имеет вид:

$P = [2, 3, 12, 8, 4, 3, 2, 6, 9, 5, 6, 10, 14, 15].$

Вершины в коде Прюфера могут повторяться, более того, в коде Прюфера может быть только одна вершина.

Так, если номер центральной вершины звезды S_4 равен 5, то код состоит из трех одинаковых цифр — 555.



Распаковка кода Прюфера

1. Находим общее количество вершин (количество вершин в коде + 2);
2. Находим висячие вершины (т.е. которые не входят в код);
3. Находим висячую вершину с минимальным номером и соединяем ее с первой неиспользованной вершиной в коде. Если выбранная вершина не встречается далее в последовательности кода, записываем ее вершину в последовательности листьев;
4. Повторяем пункт 3), до использования всех листьев;
5. Последнюю вершину кода соединяем с листом с максимальным номером.

Пример. Построить дерево, соответствующее коду Прюфера:

$P = [5, 6, 7, 8, 6, 10, 14, 15, 11, 10, 6, 7, 8, 12].$

- $A = [5, 6, 7, 8, 6, 10, 14, 15, 11, 10, 6, 7, 8, 12],$
 $N = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16],$
ребро (5, 1);
- $A = [6, 7, 8, 6, 10, 14, 15, 11, 10, 6, 7, 8, 12],$
 $N = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16],$
ребро (6, 2);

- $A = [7, 8, 6, 10, 14, 15, 11, 10, 6, 7, 8, 12]$,
 $N = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]$,
ребро (7, 3);
- $A = [8, 6, 10, 14, 15, 11, 10, 6, 7, 8, 12]$,
 $N = [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]$,
ребро (8, 4);
- $A = [6, 10, 14, 15, 11, 10, 6, 7, 8, 12]$,
 $N = [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]$,
ребро (6, 5);
- $A = [10, 14, 15, 11, 10, 6, 7, 8, 12]$,
 $N = [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]$.

Вершины 6, 7, 8 есть в A , поэтому вершину 10 (первая из списка A) соединяем с 9; получаем ребро (10, 9). Укорачиваем A и удаляем вершину 9 из N . Курсивом выделены вершины списка N , присутствующие в A .

- $A = [14, 15, 11, 10, 6, 7, 8, 12]$,
 $N = [6, 7, 8, 10, 11, 12, 13, 14, 15, 16]$.

Вершины 6, 7, 8, 10, 11, 12 из N есть в A , поэтому вершину 14 (первая из A) соединяем с вершиной 13, следующей за списком 6, 7, 8, 10, 11, 12, и получаем ребро (14, 13). Укорачиваем спереди список A , отрезая от него 14, и удаляем вершину 13 из N .

- $A = [15, 11, 10, 6, 7, 8, 12]$,
 $N = [6, 7, 8, 10, 11, 12, 14, 15, 16]$.

Аналогично получаем ребро (15, 14). Вершину 15 берем из A, вершину 14 — из N.

- $A = [11, 10, 6, 7, 8, 12]$,
 $N = [6, 7, 8, 10, 11, 12, 15, 16]$,
ребро (11, 15);
- $A = [10, 6, 7, 8, 12]$,
 $N = [6, 7, 8, 10, 11, 12, 16]$,
ребро (10, 11);
- $A = [6, 7, 8, 12]$,
 $N = [6, 7, 8, 10, 12, 16]$,
ребро (6, 10);
- $A = [7, 8, 12]$,
 $N = [6, 7, 8, 12, 16]$,
ребро (7, 6);
- $A = [8, 12]$,
 $N = [7, 8, 12, 16]$,
ребро (8, 7);

- $A = [12]$,
 $N = [8, 12, 16]$,
 ребро $(12, 8)$.

На последнем этапе получаем ребро, образованное двумя вершинами из N :

$A = []$, $N = [12, 16]$, ребро $(12, 16)$.

Дерево, закодированное по Прюферу, — свободное, т.е. оно не имеет корня. Оно может быть изображено, например, в виде, показанном на рис.1 или на рис.2.

В последнем случае в дереве искусственно выделен корень 6. Полученное дерево имеет 6 ярусов

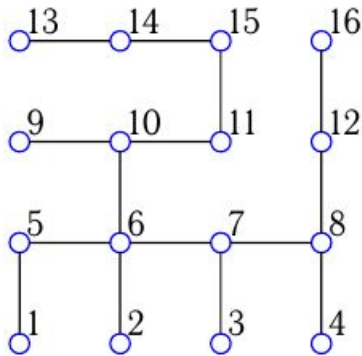


рис.1

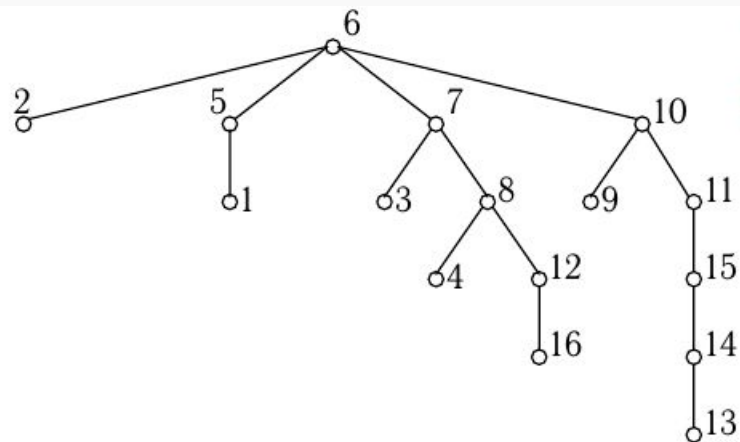


рис.2.

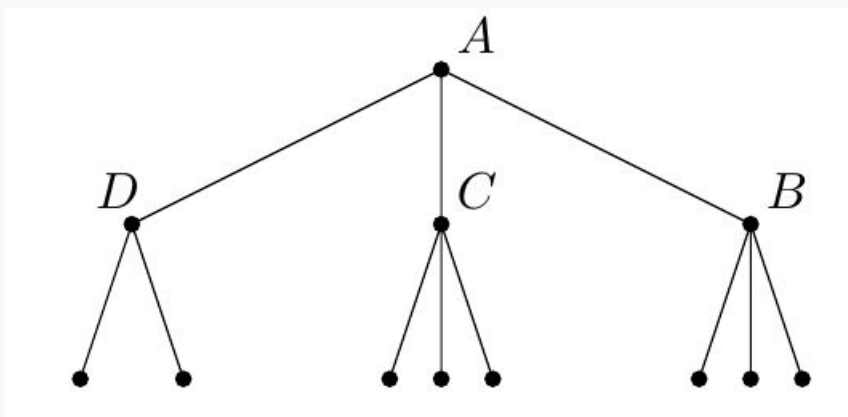
Кодировка Гапта

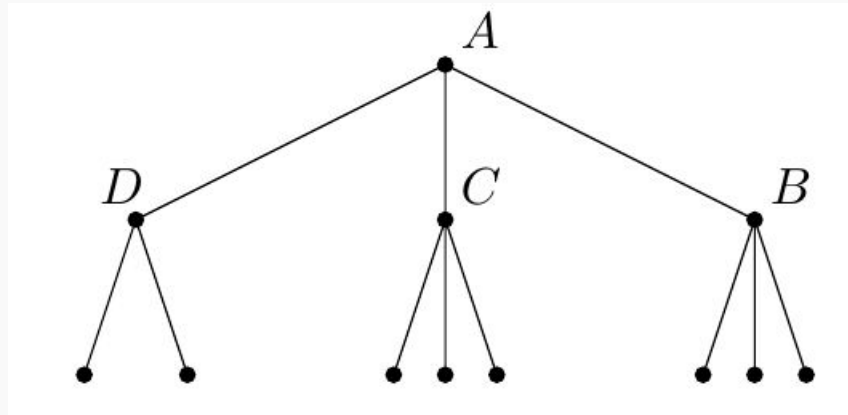
Для деревьев типа 2–3, т.е. деревьев, каждая не концевая вершина которых имеет по 2 или 3 сына, применяется код Гапта.

Без какого либо изменения алгоритма этот код обобщается и на более сложные случаи. Дерево не обязательно должно быть помечено.

Кодировка Гапта (в отличие от кодировки Прюфера) не сохраняет информацию об именах вершин.

Пример. Найти код Гапта дерева





Решение. Выберем направление обхода дерева. Пусть код состоит из числа сыновей каждой вершины дерева при обходе дерева слева направо, снизу вверх. Висячие вершины (их степень равна 1) не имеют сыновей, поэтому в код дерева порядка n , имеющего n_0 висячих вершин, войдет $n - n_0$ чисел.

Для дерева на рис. Кодировка должна содержать $12 - 8 = 4$ числа. Начнем кодировку с самой верхней вершины — А. Она имеет три сына — В, С, D. Следовательно, заносим в код число 3:

[— — —3].

Переходим на следующий ярус. Самая правая вершина, В, имеет три сына. Заносим в код число 3:

[— — 3, 3].

Продолжая далее, окончательно получаем код [2, 3, 3, 3]

Распаковка кода Гапта

Пример. По заданному коду Гапта [3, 1, 1, 1, 1, 4, 2, 1, 2, 3, 3, 3], построить дерево.

Решение. Построение начинается с корня. От корня, согласно последнему числу кода, идет три дуги к трем вершинам сыновьям 2-го яруса.

Рассматривая следующие с конца три числа кода, выясняем, что от этих вершин идет 3, 3 и 2 дуги соответственно. Изображаем эту часть дерева и продолжаем строить следующий ярус.

В результате получаем:

