

Тема 8. Гамильтоновы графы

8.1. Определение гамильтонова графа и достаточные условия гамильтоновости

Определение. Если граф имеет простой цикл, содержащий все вершины графа по одному разу, то такой цикл называется *гамильтоновым циклом*, а граф называется *гамильтоновым графом*. Если граф имеет простую цепь, содержащую все вершины графа по одному разу, то такая цепь называется *гамильтоновой цепью*, а граф называется *полугамильтоновым графом*.

Следующая теорема достаточное условие гамильтоновости графа. Необходимые условия пока не найдены.

Теорема Дирака. Если в графе $G(V, E)$ с $p \geq 3$ вершинами $\forall v \in V d(v) \geq p/2$, то граф G является гамильтоновым.

Доказательство. Пусть $V = \{v_1, \dots, v_p\}$. Заметим сначала, что любой граф можно превратить в гамильтонов добавлением в него p дополнительных вершин степени 2: вместе с дополнительной вершиной u_i , где $i = 1, \dots, p - 1$ добавляются ребра (v_i, u_i) и (u_i, v_{i+1}) , вместе с вершиной u_p – ребра (v_p, u_p) и (u_p, v_1) . Тогда гамильтонов цикл имеет вид:

$$v_1 u_1 v_2 u_2 \dots v_p u_p v_1.$$

Покажем, что можно построить гамильтонов цикл в исходном графе без дополнительной вершины u_1 . Возможны три случая:

- 1) вершины v_1 и v_2 смежны, тогда участок цикла $v_1 u_1 v_2$ можно заменить на $v_1 v_2$;
- 2) вершины v_1 и v_2 не смежны, но найдется пара вершин v_j, v_{j+1} таких, что v_j смежна с v_1 , а v_{j+1} смежна с v_2 . Тогда участок цикла $v_2 u_2 \dots v_j$ можно «развернуть», исключить дополнительные вершины u_1 и u_j , и получить цикл:

$$v_1 v_j u_{j-1} v_{j-1} \dots u_2 v_2 v_{j+1} u_{j+1} \dots v_p u_p v_1;$$

3) вершины v_1 и v_2 не смежны, и для любой вершины v_j , смежной с v_1 , вершина v_{j+1} не смежна с v_2 . Тогда количество вершин, смежных с v_1 (или $d(v_1)$), не превышает количество вершин, не смежных с v_2 (или $p - d(v_2) - 1$ – из общего количества вершин в исходном графе удаляются все вершины, смежные с v_2 , и сама вершина v_2). Отсюда имеем

$$d(v_1) \leq p - d(v_2) - 1.$$

Учитывая, что по условию теоремы $\forall v \in V : d(v) \geq p/2$ получаем

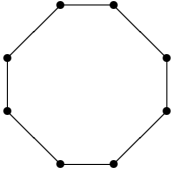
$$p = p/2 + p/2 \leq d(v_1) + d(v_2) \leq p - 1.$$

Значит, исходное предположение было неверно, и найдется пара вершин v_j, v_{j+1} таких, что v_j смежна с v_1 , а v_{j+1} смежна с v_2 . Эта ситуация рассмотрена в 2).

Значит, гамильтонов цикл можно построить и без использования дополнительной вершины u_1 . Проведя аналогичные рассуждения для всех дополнительных вершин, получим гамильтонов цикл в исходном графе. Теорема доказана.

Обратное неверно, т. е. условие теоремы не является необходимым для существования гамильтонова цикла.

Пример. $p = 8, \forall v \in V d(v) = 2$, граф является гамильтоновым.



8.2. Поиск гамильтонова цикла

Не существует алгоритмов получения гамильтонова цикла, эффективных в вычислительном смысле, то есть требующих полиномиального времени.

Алгоритм перебора Робертса и Флореса (основан на стратегии обхода «в глубину»)

Шаг 1. Составляем для всех вершин v списки смежности $L(v)$. Выбираем стартовую вершину x_1 , полагаем $i = 1$.

Шаг 2. Если $i = p$ и $\{x_p, x_1\} \in E$, то $x_1x_2 \dots x_px_1$ – гамильтонов цикл, конец. Если $i = 0$, то гамильтонова цикла нет, конец.

Шаг 3. Выбираем очередную вершину x_i , не входящую в путь $x_0x_1 \dots x_{i-1}$, из списка смежности $L(x_{i-1})$. Если такой вершины нет, то полагаем $i = i - 1$ (возвращаемся на шаг назад) и повторяем шаг 3. Иначе идем на шаг 2.

Конец.

8.3. Задача коммивояжера

Рассмотрим следующую задачу. Имеется p городов, расстояния между которыми известны. Коммивояжер должен посетить все p городов по одному разу, вернувшись в тот, с которого начал. Требуется найти такой маршрут движения, при котором суммарное пройденное расстояние будет минимальным. Очевидно, что задача сводится к отысканию кратчайшего гамильтонова цикла в полном графе. Рассмотрим методы решения этой задачи.

1. Алгоритм полного перебора. Считая, что коммивояжер всегда выезжает из одного и того же города, сгенерировать $p - 1!$ перестановок остальных вершин и для каждой перестановки вычислить длину пути, выбрав затем из них минимальную. Большая трудоемкость.

2. Методы сокращения перебора. В большинстве случаев методы сокращения перебора находят решение за меньшее время, чем алгоритм полного перебора. Однако, для любого

такого метода найдется пример, когда перебор не сокращается, т. е. в конечном случае приходится делать полный перебор.

3. Приближенные методы. Эти методы находят приближенное решение, т. е. в данном случае гамильтонов цикл, близкий к кратчайшему, за короткое время. Приближенные методы отличаются друг от друга трудоемкостью и качеством найденного решения.

8.4. Метод ветвей и границ для решения задачи коммивояжера

Суть идеи метода ветвей и границ состоит в следующем. Все перебираемые варианты разбиваются на классы. Для каждого класса находится оценка снизу всех решений этого класса. Если оценка превышает стоимость ранее найденного решения, все варианты класса отбрасываются.

В качестве первоначальной оценки решения можно взять стоимость случайного гамильтонова цикла, либо найти решение любым эвристическим методом.

Заметим, что если вычесть одно и то же число из элементов одной строки, то стоимость любого пути коммивояжера уменьшится на это число, т. к. элементы i -й строки можно интерпретировать как стоимости въезда в i -й город из остальных городов, а в любой город коммивояжер въезжает ровно один раз. Аналогичные рассуждения можно провести для столбца: вычитание одного и того же числа из элементов столбца приводит к уменьшению стоимости пути коммивояжера на это число, т. к. элементы i -го столбца можно интерпретировать как стоимости переезда из i -го города в остальные города, а из любого города коммивояжер выезжает ровно один раз.

Таким образом, если вычесть из всех элементов каждой строки минимальный элемент этой строки, а затем из элементов каждого столбца – минимальный элемент этого столбца, и сложить все эти числа, то мы получим оценку снизу стоимости пути коммивояжера. Эта операция называется *приведением* матрицы. В результате приведения матрицы в ней появляются нулевые элементы. Предполагаем, что включение в путь именно таких элементов приведет к оптимальному решению.

Каждая задача, начиная с исходной, разбивается на две подзадачи: варианты, включающие некоторую дугу $\{i, j\}$ нулевого веса и варианты, эту дугу не включающие. В первом случае, т. к. коммивояжер выезжает из города i и въезжает в город j , строка i и столбец j могут быть удалены из матрицы, что сокращает размерность задачи. После этого можно снова провести приведение матрицы. Во втором случае, т. к. проезд из города i в город j запрещен, элемент матрицы (i, j) полагается равным бесконечности. К полученной матрице опять можно применить операцию приведения, но имеет смысл проводить ее только для строки i и столбца j .

Полученные задачи опять делятся на две подзадачи, и т. д. Как только оценка снизу вариантов какого-либо класса превышает стоимость уже найденного наилучшего на данный момент решения, все эти варианты отбрасываются. Если же этого не происходит, в

результате мы получаем гамильтонов цикл меньшей стоимости, чем текущее оптимальное решение, т. е. новое оптимальное решение. Алгоритм заканчивает работу, когда все варианты либо отброшены, либо доведены до получения гамильтонова цикла.

Алгоритм ветвей и границ

Шаг 1. Строим произвольный гамильтонов цикл μ^* . Его вес - оценка сверху стоимости оптимального решения \bar{C} . Запоминаем цикл и его вес. Приводим матрицу весов W . Оценку снизу $\underline{C}(W)$ полагаем равной сумме минимальных элементов в строках и столбцах. Список задач состоит из одной задачи – приведенной матрицы весов. Маршрут коммивояжера $\mu(W)$ пуст.

Шаг 2. Если список задач пуст, идем на конец. Иначе выбираем из списка матрицу весов M .

Шаг 3. Для каждого нуля матрицы находим его *индекс* – минимальную цену, которую придется заплатить, если исключить дугу нулевой стоимости $\{i, j\}$:

$$\delta(i, j) = \min_{k \neq j} M(i, k) + \min_{k \neq i} M(k, j).$$

Шаг 4. Выбираем нулевой элемент $\{i, j\}$ с максимальным индексом, меньшим бесконечности. Если дуга $\{i, j\}$ образует цикл с уже включенными в маршрут коммивояжера дугами, полагаем $M(i, j) = \infty$, вычитаем из i -й строки, а затем из j -го столбца минимальные элементы, добавляем их сумму к $\underline{C}(M)$ и идем на шаг 3.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу $\{i, j\}$ в цикл (матрица M') и не включающую (матрица M''). Иначе получаем одну подзадачу M' .

Шаг 6. Строим матрицу M' : вычеркиваем из M i -ю строку и j -й столбец, приводим матрицу. Добавляя к $\underline{C}(M)$ сумму минимальных элементов строк и столбцов, получаем $\underline{C}(M')$. Добавляем дугу $\{i, j\}$ в маршрут коммивояжера $\mu(M)$, получаем маршрут $\mu(M')$.

Шаг 7. Если матрица M' имеет размерность 2×2 и $\underline{C}(M') < \bar{C}$, то возможно единственное решение. В маршрут коммивояжера включаем либо дуги, соответствующие главной диагонали, либо соответствующие побочной, в зависимости от того, какие именно дуги образуют гамильтонов цикл с уже включенными в маршрут $\mu(M')$ дугами. Упорядочиваем дуги $\mu(M')$ по циклу. Добавляем сумму выбранных элементов к $\underline{C}(M)$. Получен очередной маршрут коммивояжера $\mu(M')$; если его вес $\underline{C}(M') < \bar{C}$, полагаем $\bar{C} = \underline{C}(M')$ и $\mu^* = \mu(M')$ и удаляем из списка все задачи W , для которых $\underline{C}(W) \geq \bar{C}$. Если матрица M' имеет размерность больше 2×2 и $\underline{C}(M') < \bar{C}$, добавляем матрицу M' в список задач.

Шаг 8. Строим матрицу M'' : полагаем $M(i, j) = \infty$, вычитаем из i -й строки, а затем из j -го столбца минимальные элементы, добавив их сумму к $\underline{C}(M)$, получаем $\underline{C}(M'')$. Если $\underline{C}(M'') < \bar{C}$, добавляем матрицу M'' в список задач. Идем на шаг 2.

Конец. Получен оптимальный маршрут коммивояжера μ^* стоимости \bar{C} .

Пример. Рассмотрим граф $G(V, E)$ с матрицей весов W .

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>		8	4	6	12	3
<i>b</i>	7		13	9	1	7
<i>c</i>	2	13		16	11	6
<i>d</i>	5	7	18		3	14
<i>e</i>	10	2	12	4		13
<i>f</i>	2	8	3	11	10	

Шаг 1. Строим произвольный цикл $abcdefa$, его вес считаем верхней границей качества решения

$$\bar{C} = 8 + 13 + 16 + 3 + 13 + 2 = 55.$$

Выбираем минимальный элемент в каждой строке, записываем его справа.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	
<i>a</i>		8	4	6	12	3	3
<i>b</i>	7		13	9	1	7	1
<i>c</i>	2	13		16	11	6	2
<i>d</i>	5	7	18		3	14	3
<i>e</i>	10	2	12	4		13	2
<i>f</i>	2	8	3	11	10		2

Вычитаем из элементов каждой строки соответствующий минимальный элемент. Выбираем минимальный элемент в каждом столбце, записываем его снизу.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>		5	1	3	9	0
<i>b</i>	6		12	8	0	6
<i>c</i>	0	11		14	9	4
<i>d</i>	3	5	16		0	11
<i>e</i>	8	0	10	2		11
<i>f</i>	0	6	1	9	8	
	0	0	1	2	0	0

Вычитаем из элементов каждого столбца соответствующий минимальный элемент. Обозначим приведенную матрицу весов через A .

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>		5	0	1	9	0
<i>b</i>	6		11	6	0	6
<i>c</i>	0	11		12	9	4
<i>d</i>	3	5	15		0	11
<i>e</i>	8	0	9	0		11
<i>f</i>	0	6	0	7	8	

$A =$

Подсчитаем нижнюю границу качества решения для A , просуммировав все минимальные элементы.

$$\underline{C}(A) = 3 + 1 + 2 + 3 + 2 + 2 + 0 + 0 + 1 + 2 + 0 + 0 = 16.$$

Добавляем матрицу A в список задач. Маршрут коммивояжера $\mu(A) = \emptyset$.

Шаг 2. Список задач не пуст, выбираем из него матрицу A .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы A

$$\delta(a, c) = 0 + 0 = 0;$$

$$\delta(a, f) = 0 + 4 = 4;$$

$$\delta(b, e) = 6 + 0 = 6;$$

$$\delta(c, a) = 4 + 0 = 4;$$

$$\delta(d, e) = 3 + 0 = 3;$$

$$\delta(e, b) = 0 + 5 = 5;$$

$$\delta(e, d) = 0 + 1 = 1;$$

$$\delta(f, a) = 0 + 0 = 0;$$

$$\delta(f, c) = 0 + 0 = 0.$$

Итак, матрица принимает вид (в скобках выписаны индексы)

	a	b	c	d	e	f
a		5	0(0)	1	9	0(4)
b	6		11	6	0(6)	6
c	0(4)	11		12	9	4
d	3	5	15		0(3)	11
e	8	0(5)	9	0(1)		11
f	0(0)	6	0(0)	7	8	

Шаг 4. Выбираем нулевой элемент (b, e) с максимальным индексом 6. Дуга (b, e) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (b, e) в цикл (матрица B) и не включающую (матрица C).

Шаг 6. Строим матрицу B : вычеркиваем из A строку b и столбец e , получаем

	a	b	c	d	f
a		5	0	1	0
c	0	11		12	4
d	3	5	15		11
e	8	0	9	0	11
f	0	6	0	7	

Приводим матрицу, выписываем справа минимальные элементы строк, вычитаем из строк их минимальные элементы. После этого минимальные элементы столбцов равны нулю.

$$B = \begin{array}{c} \begin{array}{ccccc} & a & b & c & d & f \\ a & & 5 & 0 & 1 & 0 \\ c & 0 & 11 & & 12 & 4 \\ d & 3 & 5 & 15 & & 11 \\ e & 8 & 0 & 9 & 0 & 11 \\ f & 0 & 6 & 0 & 7 & \end{array} & \begin{array}{c} 0 \\ 0 \\ 3 \\ 0 \\ 0 \end{array} \end{array}$$

$$\begin{array}{c} \begin{array}{ccccc} & a & b & c & d & f \\ a & & 5 & 0 & 1 & 0 \\ c & 0 & 11 & & 12 & 4 \\ d & 0 & 2 & 12 & & 8 \\ e & 8 & 0 & 9 & 0 & 11 \\ f & 0 & 6 & 0 & 7 & \end{array} \end{array}$$

Добавив сумму минимальных элементов к $\underline{C}(A)$, получаем $\underline{C}()$.

$$\underline{C}(B) = \underline{C}(A) + 3 = 16 + 3 = 19.$$

Шаг 7. Матрица B имеет размерность 5×5 , и $\underline{C}(B) < \overline{C}$, добавляем матрицу B в список задач.

Шаг 8. Строим матрицу C : полагаем $A(b, e) = \infty$ (клетка (b, e) теперь пустая), вычитаем из строки b , а затем из столбца e минимальные элементы.

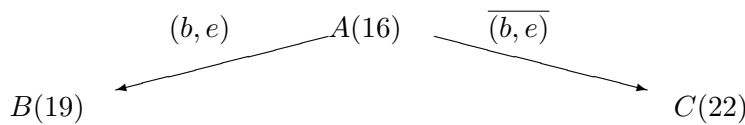
$$C = \begin{array}{c} \begin{array}{ccccc} & a & b & c & d & e & f \\ a & & 5 & 0 & 1 & 9 & 0 \\ b & 6 & & 11 & 6 & & 6 \\ c & 0 & 11 & & 12 & 9 & 4 \\ d & 3 & 5 & 15 & & 0 & 11 \\ e & 8 & 0 & 9 & 0 & & 11 \\ f & 0 & 6 & 0 & 7 & 8 & \end{array} \end{array}$$

$$C = \begin{array}{c} \begin{array}{ccccc} & a & b & c & d & e & f \\ a & & 5 & 0 & 1 & 9 & 0 \\ b & 0 & & 5 & 0 & & 0 \\ c & 0 & 11 & & 12 & 9 & 4 \\ d & 3 & 5 & 15 & & 0 & 11 \\ e & 8 & 0 & 9 & 0 & & 11 \\ f & 0 & 6 & 0 & 7 & 8 & \end{array} \end{array}$$

Добавив их сумму к $\underline{C}(A)$, получаем $\underline{C}(C)$.

$$\underline{C}(C) = \underline{C}(A) + 6 = 16 + 6 = 22.$$

Поскольку $\underline{C}(C) < \overline{C}$, добавляем матрицу C в список задач. В итоге дерево решений имеет вид (в скобках указана нижняя граница решения).



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу B .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы B

$$B = \begin{array}{c} \begin{array}{ccccc} & a & b & c & d & f \\ a & & 5 & 0(0) & 1 & 0(4) \\ c & 0(4) & 11 & & 12 & 4 \\ d & 0(2) & 2 & 12 & & 8 \\ e & 8 & 0(2) & 9 & 0(1) & 11 \\ f & 0(0) & 6 & 0(0) & 7 & \end{array} \end{array}$$

Шаг 4. Выбираем нулевой элемент (a, f) с максимальным индексом 4. Дуга (a, f) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу $a, f)$ в цикл (матрица D) и не включающую (матрица E).

Шаг 6. Строим матрицу D : вычеркиваем из B строку a и столбец f , получаем

$$D = \begin{array}{c|cccc} & a & b & c & d \\ \hline c & 0 & 11 & & 12 \\ d & 0 & 2 & 12 & \\ e & 8 & 0 & 9 & 0 \\ f & 0 & 6 & 0 & 7 \end{array}$$

Минимальные элементы строк и столбцов – нули, приводить матрицу не требуется,

$$\underline{C}(D) = \underline{C}(B) = 19.$$

Добавляем дугу (a, f) в маршрут $\mu(B)$, получаем маршрут $\mu(D) = \{(b, e), (a, f)\}$.

Шаг 7. Матрица D имеет размерность 4×4 , и $\underline{C}(E) < \bar{C}$, добавляем матрицу E в список задач.

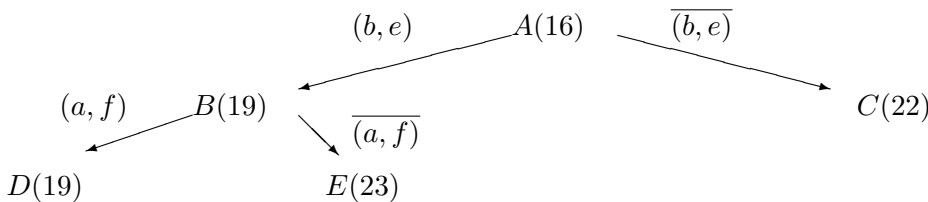
Шаг 8. Строим матрицу E : полагаем $B(a, f) = \infty$, вычитаем из строки a , а затем из столбца f минимальные элементы.

$$E = \begin{array}{c|ccccc} & a & b & c & d & f \\ \hline a & & 5 & 0 & 1 & \\ c & 0 & 11 & & 12 & 4 \\ d & 0 & 2 & 12 & & 8 \\ e & 8 & 0 & 9 & 0 & 11 \\ f & 0 & 6 & 0 & 7 & \end{array} \quad E = \begin{array}{c|ccccc} & a & b & c & d & f \\ \hline a & & 5 & 0 & 1 & \\ c & 0 & 11 & & 12 & 0 \\ d & 0 & 2 & 12 & & 4 \\ e & 8 & 0 & 9 & 0 & 7 \\ f & 0 & 6 & 0 & 7 & \end{array}$$

Добавив их сумму к $\underline{C}(B)$, получаем $\underline{C}(E)$.

$$\underline{C}(E) = \underline{C}(B) + 4 = 19 + 4 = 23.$$

Поскольку $\underline{C}(E) < \bar{C}$, добавляем матрицу E в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу D .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы D .

$$D = \begin{array}{c|cccc} & a & b & c & d \\ \hline c & 0(11) & 11 & & 12 \\ d & 0(2) & 2 & 12 & \\ e & 8 & 0(2) & 9 & 0(7) \\ f & 0(0) & 6 & 0(9) & 7 \end{array}$$

Шаг 4. Выбираем нулевой элемент (c, a) с максимальным индексом 11. Дуга (c, a) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (c, a) в цикл (матрица F) и не включающую (матрица G).

Шаг 6. Строим матрицу F : вычеркиваем из D строку c и столбец a . Указываем справа нулевые элементы, затем вычитаем их из элементов соответствующих строк. Приводим матрицу.

$$F = \begin{array}{c|ccc|c} & b & c & d & \\ \hline d & 2 & 12 & & 2 \\ e & 0 & 9 & 0 & 0 \\ f & 6 & 0 & 7 & 0 \end{array} \quad F = \begin{array}{c|ccc} & b & c & d \\ \hline d & 0 & 10 & \\ e & 0 & 9 & 0 \\ f & 6 & 0 & 7 \end{array}$$

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(F) = \underline{C}(D) = 19 + 2 = 21.$$

Добавляем дугу (c, a) в маршрут $\mu(D)$, получаем маршрут $\mu(F) = \{(b, e), (a, f), (c, a)\}$.

Шаг 7. Матрица F имеет размерность 3×3 , и $\underline{C}(F) < \bar{C}$, добавляем матрицу E в список задач.

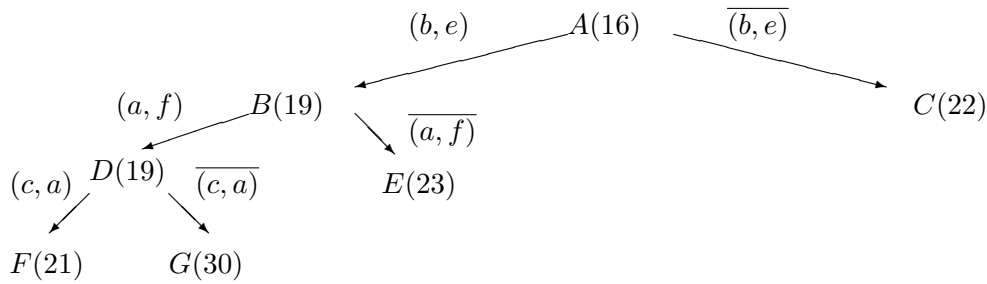
Шаг 8. Строим матрицу G : полагаем $D(c, a) = \infty$, вычитаем из строки c , а затем из столбца a минимальные элементы.

$$G = \begin{array}{c|cccc} & a & b & c & d \\ \hline c & & 11 & & 12 \\ d & 0 & 2 & 12 & \\ e & 8 & 0 & 9 & 0 \\ f & 0 & 6 & 0 & 7 \end{array} \quad G = \begin{array}{c|cccc} & a & b & c & d \\ \hline c & & 0 & & 1 \\ d & 0 & 2 & 12 & \\ e & 8 & 0 & 9 & 0 \\ f & 0 & 6 & 0 & 7 \end{array}$$

Добавив их сумму к $\underline{C}(D)$, получаем $\underline{C}(G)$.

$$\underline{C}(G) = \underline{C}(D) + 11 = 19 + 11 = 30.$$

Поскольку $\underline{C}(G) < \bar{C}$, добавляем матрицу G в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу F .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы F .

$$F = \begin{array}{c|ccc} & b & c & d \\ \hline d & 0(10) & 10 & \\ e & 0(0) & 9 & 0(7) \\ f & 6 & 0(9) & 7 \end{array}$$

Шаг 4. Выбираем нулевой элемент (d, b) с максимальным индексом 10. Дуга (d, b) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (d, b) в цикл (матрица H) и не включающую (матрица I).

Шаг 6. Строим матрицу H : вычеркиваем из F строку d и столбец b . Каждая строка и каждый столбец содержит нули, матрица приведена.

$$H = \begin{array}{c|cc} & c & d \\ \hline e & 9 & 0 \\ f & 0 & 7 \end{array}$$

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(H) = \underline{C}(F) = 21.$$

Добавляем дугу (d, b) в маршрут $\mu(F)$, получаем маршрут $\mu(H) = \{(b, e), (a, f), (c, a), (d, b)\}$.

Шаг 7. Матрица H имеет размерность 2×2 , и $\underline{C}(H) < \bar{C}$. Дуга (e, d) образует цикл $edbe$ с уже включенными в маршрут коммивояжера дугами (d, b) и (b, e) . Поэтому добавляем в маршрут коммивояжера $\mu(H)$ дуги (e, c) и (f, d) и упорядочиваем его дуги по циклу:

$$\mu(H) = \{(b, e), (a, f), (c, a), (d, b), (e, c), (f, d)\} = becafdb.$$

Рассчитываем нижнюю границу качества решения, добавляя к предыдущему ее значению сумму выбранных элементов.

$$\underline{C}(H) = 21 + 9 + 7 = 37.$$

Получен очередной маршрут коммивояжера $\mu(H)$; его вес $\underline{C}(H) = 37 < \bar{C}$, полагаем $\bar{C} = 37$ и $\mu^* = becafdb$.

Шаг 8. Строим матрицу I : полагаем $F(d, b) = \infty$, вычитаем из строки d , а затем из столбца b минимальные элементы.

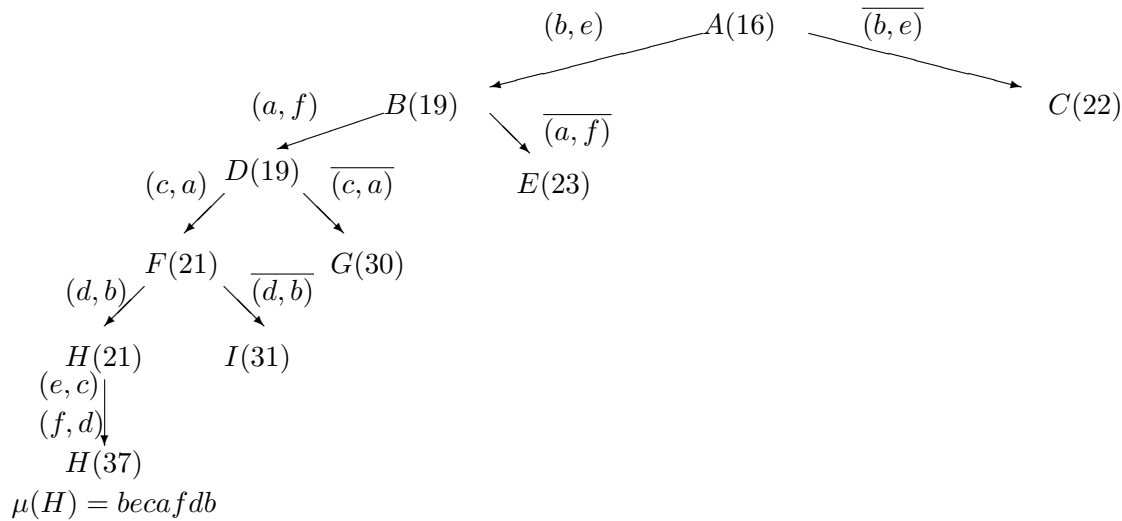
$$I = \begin{array}{c|ccc} & b & c & d \\ \hline d & & 10 & \\ \hline e & 0 & 9 & 0 \\ \hline f & 6 & 0 & 7 \\ \hline \end{array}$$

$$I = \begin{array}{c|ccc} & b & c & d \\ \hline d & & 0 & \\ \hline e & 0 & 9 & 0 \\ \hline f & 6 & 0 & 7 \\ \hline \end{array}$$

Добавив их сумму к $\underline{C}(F)$, получаем $\underline{C}(I)$.

$$\underline{C}(I) = \underline{C}(F) + 10 = 21 + 10 = 31.$$

Поскольку $\underline{C}(I) < \overline{C}$, добавляем матрицу I в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу C .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы C .

$$C = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & & 5 & 0(0) & 1 & 9 & 0(0) \\ \hline b & 0(0) & & 5 & 0(0) & & 0(0) \\ \hline c & 0(4) & 11 & & 12 & 9 & 4 \\ \hline d & 3 & 5 & 15 & & 0(11) & 11 \\ \hline e & 8 & 0(5) & 9 & 0(0) & & 11 \\ \hline f & 0(0) & 6 & 0(0) & 7 & 8 & \\ \hline \end{array}$$

Шаг 4. Выбираем нулевой элемент (d, e) с максимальным индексом 11. Дуга (d, e) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (d, e) в цикл (матрица J) и не включающую (матрица K).

Шаг 6. Строим матрицу J : вычеркиваем из C строку d и столбец e . Каждая строка и каждый столбец содержит нули, матрица приведена.

$$J = \begin{array}{c|ccccc} & a & b & c & d & f \\ \hline a & & 5 & 0 & 1 & 0 \\ b & 0 & & 5 & 0 & 0 \\ c & 0 & 11 & & 12 & 4 \\ e & 8 & 0 & 9 & 0 & 11 \\ f & 0 & 6 & 0 & 7 & \end{array}$$

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(J) = \underline{C}(C) = 22.$$

Добавляем дугу (d, e) в маршрут $\mu(C)$, получаем маршрут $\mu(I) = \{(d, e)\}$.

Шаг 7. Матрица J имеет размерность 5×5 , и $\underline{C}(I) < \bar{C}$, добавляем матрицу I в список задач.

Шаг 8. Строим матрицу K : полагаем $C(d, e) = \infty$, вычитаем из строки d , а затем из столбца e минимальные элементы.

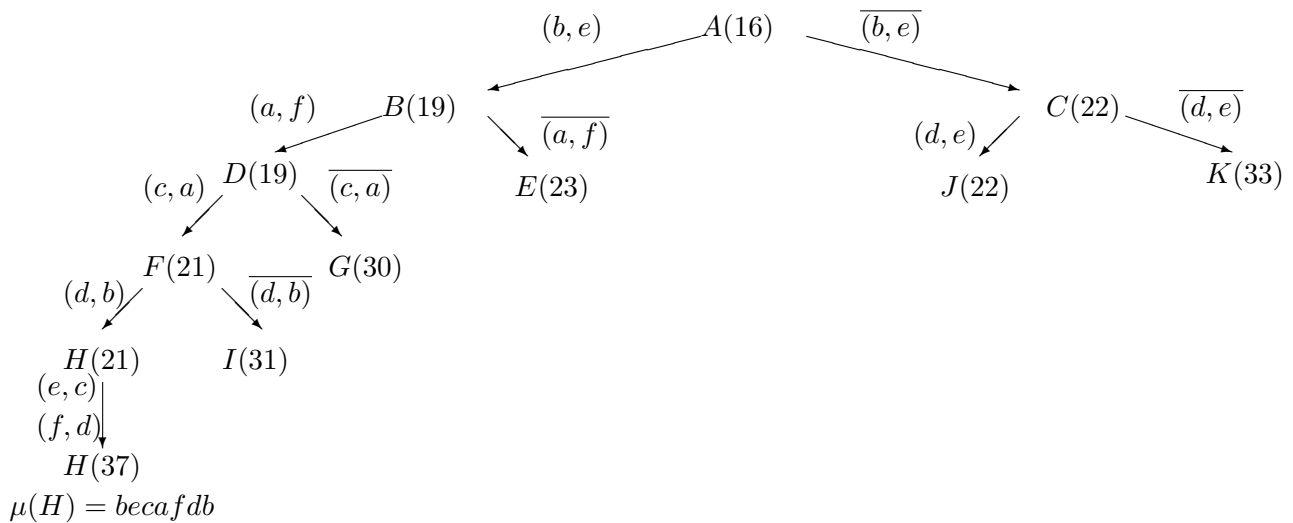
$$K = \begin{array}{c|ccccc} & a & b & c & d & e & f \\ \hline a & & 5 & 0 & 1 & 9 & 0 \\ b & 0 & & 5 & 0 & & 0 \\ c & 0 & 11 & & 12 & 9 & 4 \\ d & 3 & 5 & 15 & & & 11 \\ e & 8 & 0 & 9 & 0 & & 11 \\ f & 0 & 6 & 0 & 7 & 8 & \end{array}$$

$$K = \begin{array}{c|ccccc} & a & b & c & d & e & f \\ \hline a & & 5 & 0 & 1 & 1 & 0 \\ b & 0 & & 5 & 0 & & 0 \\ c & 0 & 11 & & 12 & 1 & 4 \\ d & 0 & 2 & 12 & & & 8 \\ e & 8 & 0 & 9 & 0 & & 11 \\ f & 0 & 6 & 0 & 7 & 0 & \end{array}$$

Добавив их сумму к $\underline{C}(C)$, получаем $\underline{C}(K)$.

$$\underline{C}(K) = \underline{C}(C) + 11 = 22 + 11 = 33.$$

Поскольку $\underline{C}(K) < \bar{C}$, добавляем матрицу K в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу J .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы J .

	a	b	c	d	f
$J =$		5	0(0)	1	0(0)
	0(0)		5	0(0)	0(0)
	0(4)	11		12	4
	8	0(5)	9	0(0)	11
	0(0)	6	0(0)	7	

Шаг 4. Выбираем нулевой элемент (e, b) с максимальным индексом 5. Дуга (e, b) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (e, b) в цикл (матрица L) и не включающую (матрица M).

Шаг 6. Строим матрицу L : вычеркиваем из J строку e и столбец b . Каждая строка и каждый столбец содержит нули, матрица приведена.

	a	c	d	f
$L =$		0	1	0
	0	5	0	0
	0		12	4
	0	6	7	

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(L) = \underline{C}(J) = 22.$$

Добавляем дугу (e, b) в маршрут $\mu(J)$, получаем маршрут $\mu(L) = \{(d, e), (e, b)\}$.

Шаг 7. Матрица L имеет размерность 4×4 , и $\underline{C}(L) < \bar{C}$, добавляем матрицу L в список задач.

Шаг 8. Строим матрицу M : полагаем $J(e, b) = \infty$, вычитаем из строки e , а затем из столбца b минимальные элементы.

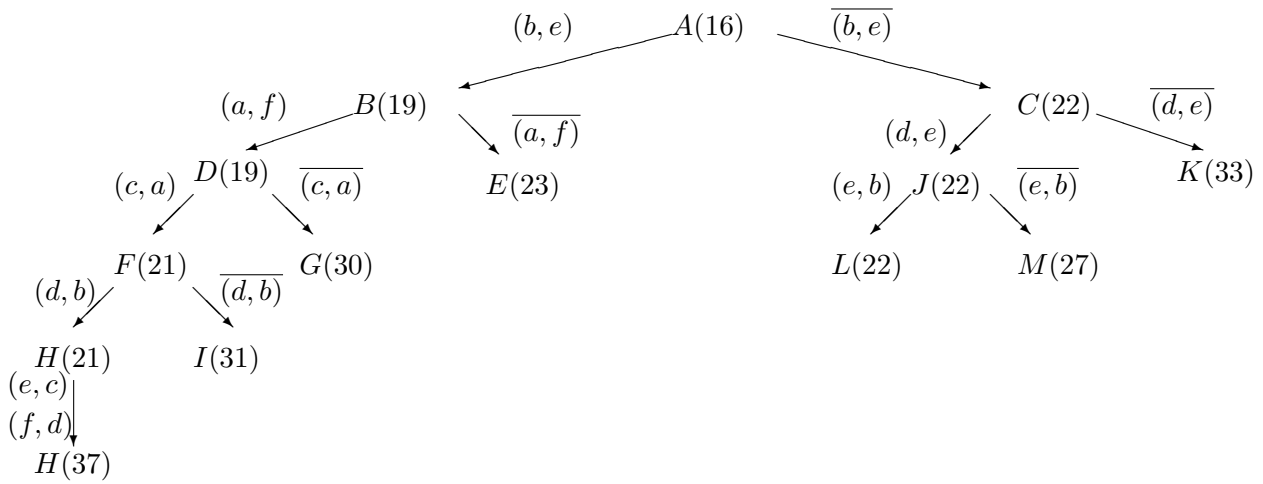
	a	b	c	d	f
$M =$		5	0	1	0
	0		5	0	0
	0	11		12	4
	8		9	0	11
	0	6	0	7	

	a	b	c	d	f
$M =$		0	0	1	0
	0		5	0	0
	0	6		12	4
	8		9	0	11
	0	1	0	7	

Добавив их сумму к $\underline{C}(J)$, получаем $\underline{C}(M)$.

$$\underline{C}(M) = \underline{C}(J) + 5 = 22 + 5 = 27.$$

Поскольку $\underline{C}(M) < \bar{C}$, добавляем матрицу M в список задач. В итоге дерево решений имеет вид.



$\mu(H) = becafdb$

Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу L .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы L .

	a	c	d	f
a		0(5)	1	0(0)
b	0(0)	5	0(1)	0(0)
c	0(4)		12	4
f	0(6)	6	7	

Шаг 4. Выбираем нулевой элемент (f, a) с максимальным индексом 6. Дуга (f, a) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (f, a) в цикл (матрица N) и не включающую (матрица O).

Шаг 6. Строим матрицу N : вычеркиваем из L строку f и столбец a . Указываем справа нулевые элементы, затем вычитаем их из элементов соответствующих строк. Приводим матрицу.

	c	d	f	
a	0	1	0	
b	5	0	0	
c		12	4	4

	c	d	f
a	0	1	0
b	5	0	0
c		8	0

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(N) = \underline{C}(L) + 4 = 22 + 4 = 26.$$

Добавляем дугу (f, a) в маршрут $\mu(L)$, получаем маршрут $\mu(N) = \{(d, e), (e, b), (f, a)\}$.

Шаг 7. Матрица L имеет размерность 3×3 , и $\underline{C}(N) < \bar{C}$, добавляем матрицу L в список задач.

Шаг 8. Строим матрицу O : полагаем $L(f, a) = \infty$, вычитаем из строки f , а затем из столбца a минимальные элементы.

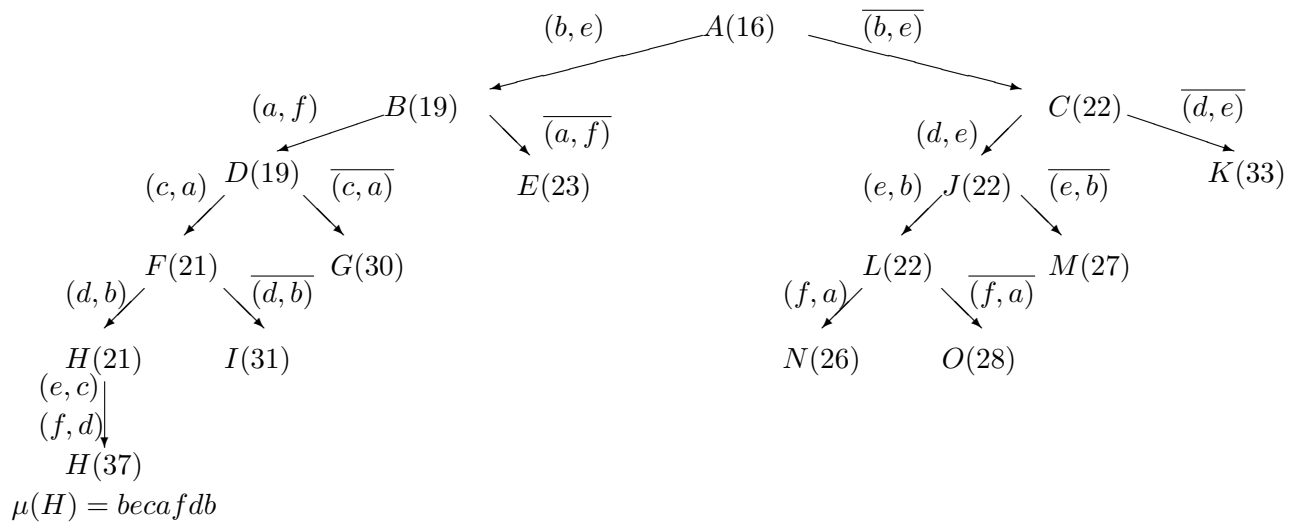
$$O = \begin{array}{c|ccc} & a & c & d & f \\ \hline a & & 0 & 1 & 0 \\ b & 0 & 5 & 0 & 0 \\ c & 0 & & 12 & 4 \\ f & & 6 & 7 & \end{array}$$

$$O = \begin{array}{c|ccc} & a & c & d & f \\ \hline a & & 0 & 1 & 0 \\ b & 0 & 5 & 0 & 0 \\ c & 0 & & 12 & 4 \\ f & & 0 & 1 & \end{array}$$

Добавив их сумму к $\underline{C}(L)$, получаем $\underline{C}(O)$.

$$\underline{C}(O) = \underline{C}(L) + 6 = 22 + 6 = 28.$$

Поскольку $\underline{C}(O) < \bar{C}$, добавляем матрицу O в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу N .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы N .

$$N = \begin{array}{c|ccc} & c & d & f \\ \hline a & 0(5) & 1 & 0(0) \\ b & 5 & 0(1) & 0(0) \\ c & & 8 & 0(8) \end{array}$$

Шаг 4. Выбираем нулевой элемент (c, f) с максимальным индексом 8. Дуга (c, f) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (c, f) в цикл (матрица P) и не включающую (матрица R).

Шаг 6. Строим матрицу P : вычеркиваем из N строку c и столбец f . Матрица приведена.

$$N = \begin{array}{c|cc} & c & d \\ \hline a & 0 & 1 \\ b & 5 & 0 \end{array}$$

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(P) = \underline{C}(N) = 26.$$

Добавляем дугу (c, f) в маршрут $\mu(N)$, получаем маршрут $\mu(P) = \{(d, e), (e, b), (f, a), (c, f)\}$.

Шаг 7. Матрица P имеет размерность 2×2 , и $\underline{C}(P) < \bar{C}$. Дуга (b, d) образует цикл $bdeb$ с уже включенными в маршрут коммивояжера дугами (d, e) и (e, b) . Поэтому добавляем в маршрут коммивояжера $\mu(P)$ дуги (b, c) и (a, d) и упорядочиваем его дуги по циклу:

$$\mu(P) = \{(d, e), (e, b), (f, a), (c, f), (b, c), (a, d)\} = debcfad.$$

Рассчитываем нижнюю границу качества решения, добавляя к предыдущему ее значению сумму выбранных элементов.

$$\underline{C}(P) = 26 + 5 + 1 = 32.$$

Получен очередной маршрут коммивояжера $\mu(P)$; его вес $\underline{C}(P) = 2 < \bar{C}$, полагаем $\bar{C} = 32$ и $\mu^* = debcfad$. Поскольку $\underline{C}(K) < \bar{C}$, удаляем K из списка задач.

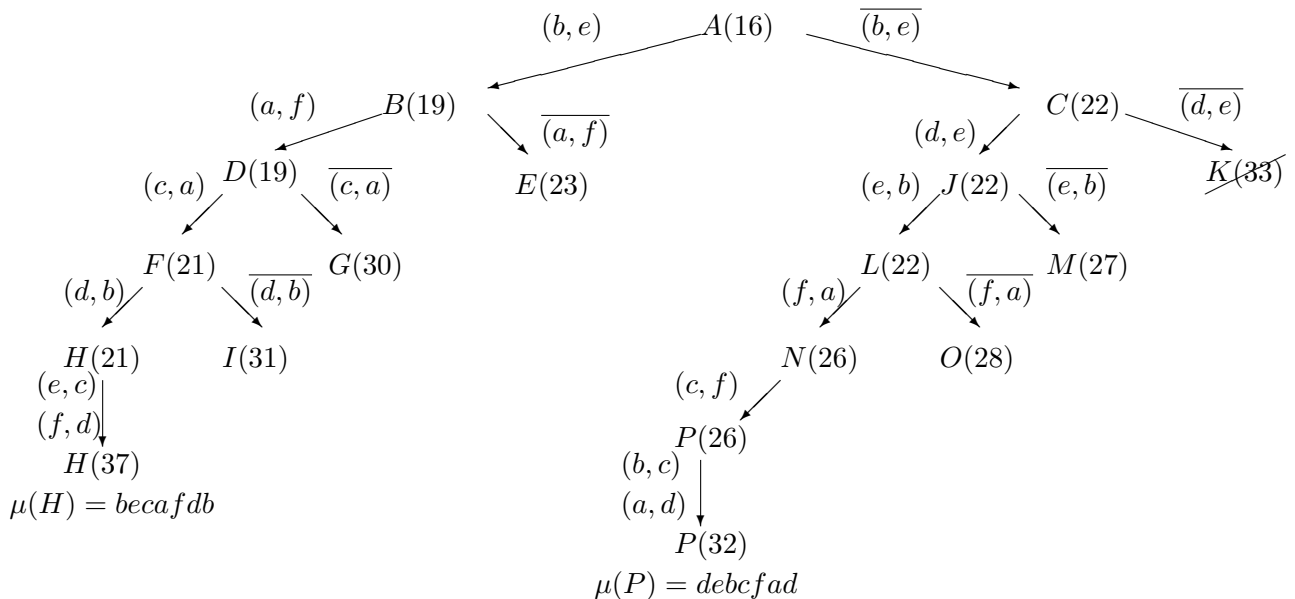
Шаг 8. Строим матрицу R : полагаем $N(c, f) = \infty$, вычитаем из строки c , а затем из столбца f минимальные элементы.

$$R = \begin{array}{c|ccc} & c & d & f \\ \hline a & 0 & 1 & 0 \\ b & 5 & 0 & 0 \\ c & & 8 & \end{array} \quad R = \begin{array}{c|ccc} & c & d & f \\ \hline a & 0 & 1 & 0 \\ b & 5 & 0 & 0 \\ c & & 0 & \end{array}$$

Добавив их сумму к $\underline{C}(N)$, получаем $\underline{C}(R)$.

$$\underline{C}(R) = \underline{C}(N) + 8 = 26 + 8 = 34.$$

Поскольку $\underline{C}(P) < \bar{C}$, не добавляем матрицу P в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу E .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы E .

	a	b	c	d	f
$E =$	a	5	0(1)	1	
	c	0(0)	11		12
	d	0(2)	2	12	
	e	8	0(2)	9	0(1)
	f	0(0)	6	0(0)	7

Шаг 4. Выбираем нулевой элемент (c, f) с максимальным индексом 4. Дуга (c, f) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (c, f) в цикл (матрица S) и не включающую (матрица T).

Шаг 6. Строим матрицу S : вычеркиваем из E строку c и столбец f . Матрица приведена.

	a	b	c	d
$S =$	a	5	0	1
	d	0	2	12
	e	8	0	9
	f	0	6	0

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(S) = \underline{C}(E) = 23.$$

Добавляем дугу (c, f) в маршрут $\mu(E)$, получаем маршрут $\mu(S) = \{(b, e), (c, f)\}$.

Шаг 7. Матрица S имеет размерность 4×4 , и $\underline{C}(S) < \bar{C}$. Добавляем ее в список задач.

Шаг 8. Строим матрицу T : полагаем $E(c, f) = \infty$, вычитаем из строки c , а затем из столбца f минимальные элементы.

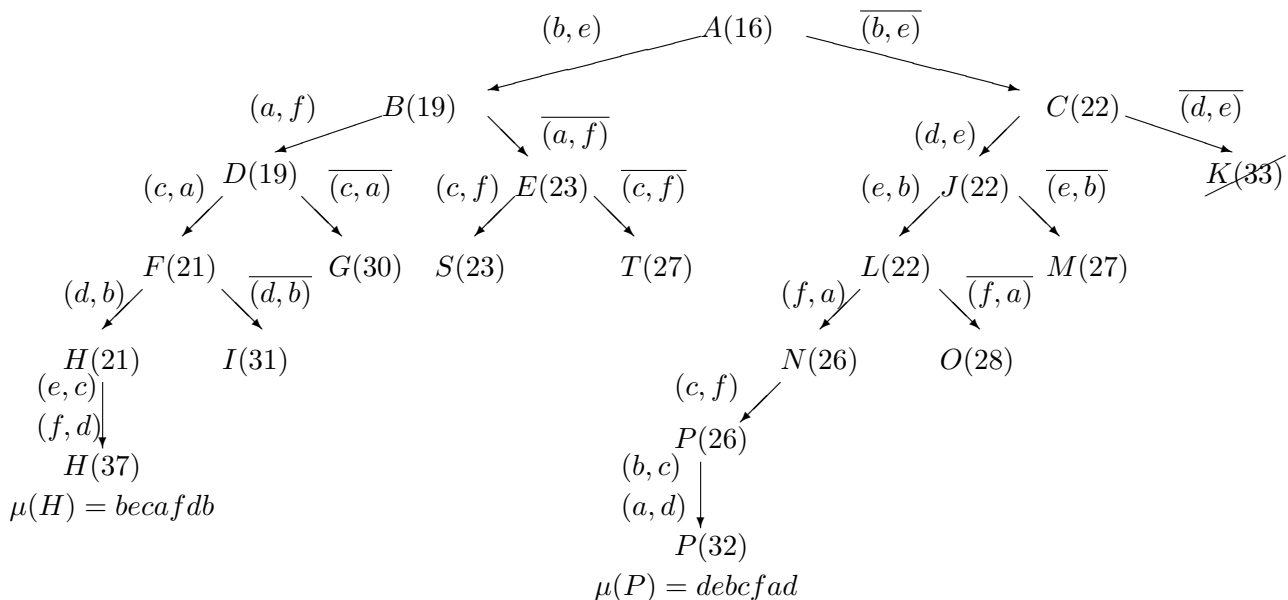
	a	b	c	d	f
$T =$	a	5	0	1	
	c	0	11		12
	d	0	2	12	4
	e	8	0	9	0
	f	0	6	0	7

	a	b	c	d	f
$T =$	a	5	0	1	
	c	0	11		12
	d	0	2	12	0
	e	8	0	9	0
	f	0	6	0	7

Добавив их сумму к $\underline{C}(E)$, получаем $\underline{C}(T)$.

$$\underline{C}(T) = \underline{C}(E) + 4 = 23 + 4 = 27.$$

Поскольку $\underline{C}(T) < \bar{C}$, не добавляем матрицу T в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу S .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы S .

	a	b	c	d
a		5	0(1)	1
d	0(2)	2	12	
e	8	0(2)	9	0(1)
f	0(0)	6	0(0)	7

Шаг 4. Выбираем нулевой элемент (e, b) с максимальным индексом 2. Дуга (e, b) образует цикл beb с уже включенными в маршрут коммивояжера дугами (b, e) . Полагаем $S(b, e) = \infty$ и вычитаем из строки e и столбца b их минимальные элементы.

	a	b	c	d
a		5	0	1
d	0	2	12	
e	8		9	0
f	0	6	0	7

	a	b	c	d
a		3	0	1
d	0	0	12	
e	8		9	0
f	0	4	0	7

Добавляем сумму этих элементов к $\underline{C}(U)$.

$$\underline{C}(U) = 23 + 2 = 25.$$

Поскольку $\underline{C}(U) < \overline{C}$, идем на шаг 3.

Шаг 3. Рассчитываем индексы нулевых элементов матрицы S .

$$S = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & 3 & 0(1) & 1 \\ d & 0(0) & 0(3) & 12 & \\ e & 8 & & 9 & 0(1) \\ f & 0(0) & 4 & 0(0) & 7 \end{array}$$

Шаг 4. Выбираем нулевой элемент (d, b) с максимальным индексом 3. Дуга (d, b) не образует цикла с уже включенными в маршрут коммивояжера дугами.

Шаг 5. Текущую задачу делим на две подзадачи: включающую дугу (d, b) в цикл (матрица U) и не включающую (матрица V).

Шаг 6. Строим матрицу U : вычеркиваем из S строку d и столбец b . Матрица приведена.

$$U = \begin{array}{c|ccc} & a & c & d \\ \hline a & & 0 & 1 \\ e & 8 & 9 & 0 \\ f & 0 & 0 & 7 \end{array}$$

Рассчитываем нижнюю границу качества решения.

$$\underline{C}(U) = \underline{C}(S) = 25.$$

Добавляем дугу (d, b) в маршрут $\mu(S)$, получаем маршрут $\mu(U) = \{(b, e), (c, f), (d, b)\}$.

Шаг 7. Матрица U имеет размерность 3×3 , и $\underline{C}(U) < \bar{C}$. Добавляем ее в список задач.

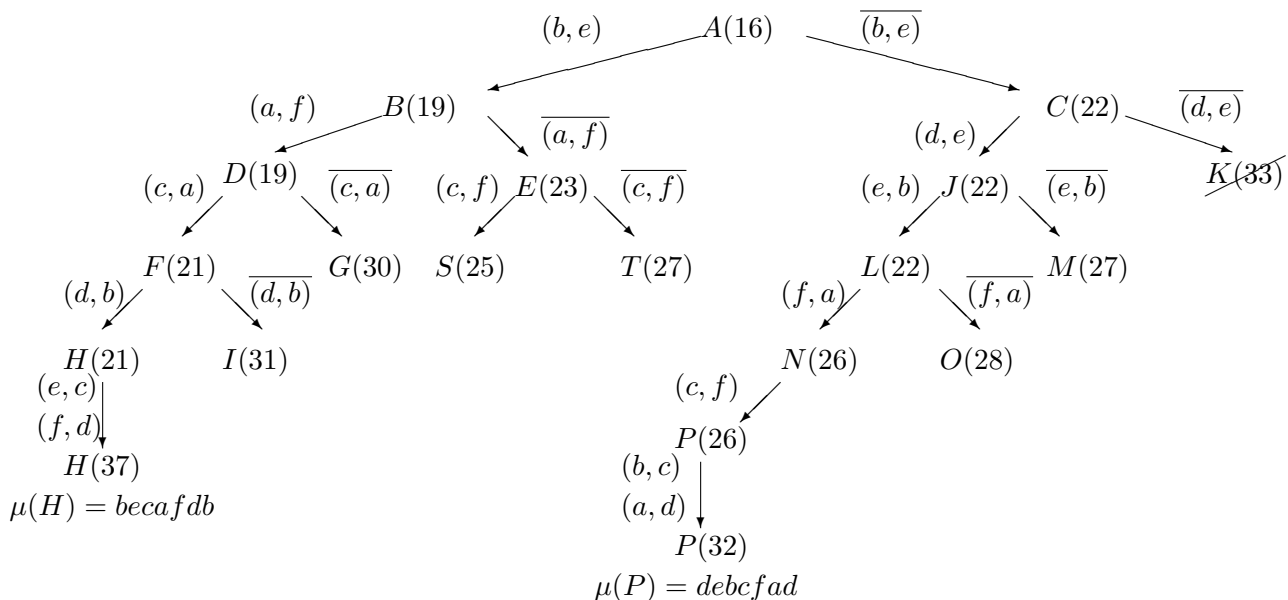
Шаг 8. Строим матрицу V : полагаем $S(d, b) = \infty$, вычитаем из строки d , а затем из столбца b минимальные элементы.

$$V = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & 3 & 0 & 1 \\ d & 0 & & 12 & \\ e & 8 & & 9 & 0 \\ f & 0 & 4 & 0 & 7 \end{array} \quad V = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & 0 & 0 & 1 \\ d & 0 & & 12 & \\ e & 8 & & 9 & 0 \\ f & 0 & 1 & 0 & 7 \end{array}$$

Добавив их сумму к $\underline{C}(S)$, получаем $\underline{C}(V)$.

$$\underline{C}(V) = \underline{C}(S) + 3 = 25 + 3 = 28.$$

Поскольку $\underline{C}(T) < \bar{C}$, добавляем матрицу T в список задач. В итоге дерево решений имеет вид.



Идем на шаг 2.

Шаг 2. Список задач не пуст, выбираем из него матрицу U .

Шаг 3. Рассчитываем индексы нулевых элементов матрицы U .

$$U = \begin{array}{c|ccc} & a & c & d \\ \hline a & & 0(1) & 1 \\ e & 8 & 9 & 0(9) \\ f & 0(8) & 0(0) & 7 \end{array}$$

Шаг 4. Выбираем нулевой элемент (e, d) с максимальным индексом 9. Дуга (e, d) образует цикл $edbe$ с уже включенными в маршрут коммивояжера дугами (b, e) и (d, b) . Полагаем $U(e, d) = \infty$ и вычитаем из строки e и столбца d их минимальные элементы.

$$U = \begin{array}{c|ccc} & a & c & d \\ \hline a & & 0 & 1 \\ e & 8 & 9 & \\ f & 0 & 0 & 7 \end{array} \quad U = \begin{array}{c|ccc} & a & c & d \\ \hline a & & 0 & 0 \\ e & 0 & 1 & \\ f & 0 & 0 & 6 \end{array}$$

Добавляем сумму этих элементов к $\underline{C}(U)$.

$$\underline{C}(U) = 25 + 9 = 34.$$

Так как $\underline{C}(U) > \bar{C}$, не рассматриваем дальше задачу U и идем на шаг 2.

8.5. Некоторые эвристические методы решения задачи коммивояжера

1) Алгоритм ближайшего соседа. На каждом шаге включаем в цепь ребро наименьшего веса из ребер, соединяющих последнюю присоединенную вершину с вершинами, еще не включенным в цепь. Когда все вершины оказываются включенными в цепь, возвращаемся к исходной вершине по единственно возможному ребру. Как правило, вес построенного цикла далек от минимального. Для любого положительного k найдется граф,

для которого вес гамильтонова цикла, построенного по этому алгоритму, в k раз больше минимального.

2) Алгоритм ближайшей вставки. Возьмем в графе некоторый цикл и на каждом шаге будем увеличивать его, включая самые близкие к нему вершины. Вес построенного цикла не более чем в 2 раза больше минимального веса.

3) Метод локальной оптимизации. Метод заключается в том, что в имеющемся гамильтоновом цикле для всех вершин $i, j : 1 \leq i < j \leq p$ делается попытка поменять местами вершины i и j . Если эта перестановка уменьшает вес цикла, результат запоминается как очередное решение. Процесс продолжается, пока происходит улучшение решения.

4) Алгоритм Эйлера. Этот метод работает, если выполняется неравенство треугольника:

$$\forall i, j, k : W(i, j) + W(j, k) \leq W(i, k).$$

Шаг 1. Строится остов минимального веса (алгоритмом Прима или Краскала).

Шаг 2. Путем дублирования каждого ребра остов превращается в эйлеров граф.

Шаг 3. В полученном графе находится эйлеров цикл (алгоритм Флери).

Шаг 4. Эйлеров цикл преобразуется в гамильтонов цикл: последовательность вершин эйлерова цикла сокращается так, чтобы каждая вершина графа в полученном цикле встречалась ровно один раз.

Покажем, что вес найденного решения R не превосходит удвоенного веса оптимального решения W^* . Для начала заметим, что вес кратчайшего остова T не превосходит W^* , поскольку при удалении любого ребра из оптимального маршрута коммивояжера получается остов веса не меньше, чем T . Вес эйлерова цикла равен $2T$. Неравенство треугольника обеспечивает оценку $R < 2T$, а значит, $R < 2T < 2W^*$.

5) Алгоритм Кристофидеса. Отличается от предыдущего логикой построения гамильтонова цикла из кратчайшего остова.

Шаг 1. Строится остов минимального веса (алгоритмом Прима или Краскала).

Шаг 2. На множестве вершин остова нечетной степени находится паросочетание минимального веса.

Шаг 3. Остов преобразуется в эйлеров граф путем присоединения ребер, соответствующих найденному паросочетанию.

Шаг 4. В полученном графе находится эйлеров цикл (алгоритм Флери).

Шаг 5. Эйлеров цикл преобразуется в гамильтонов цикл: последовательность вершин эйлерова цикла сокращается так, чтобы каждая вершина графа в полученном цикле встречалась ровно один раз.