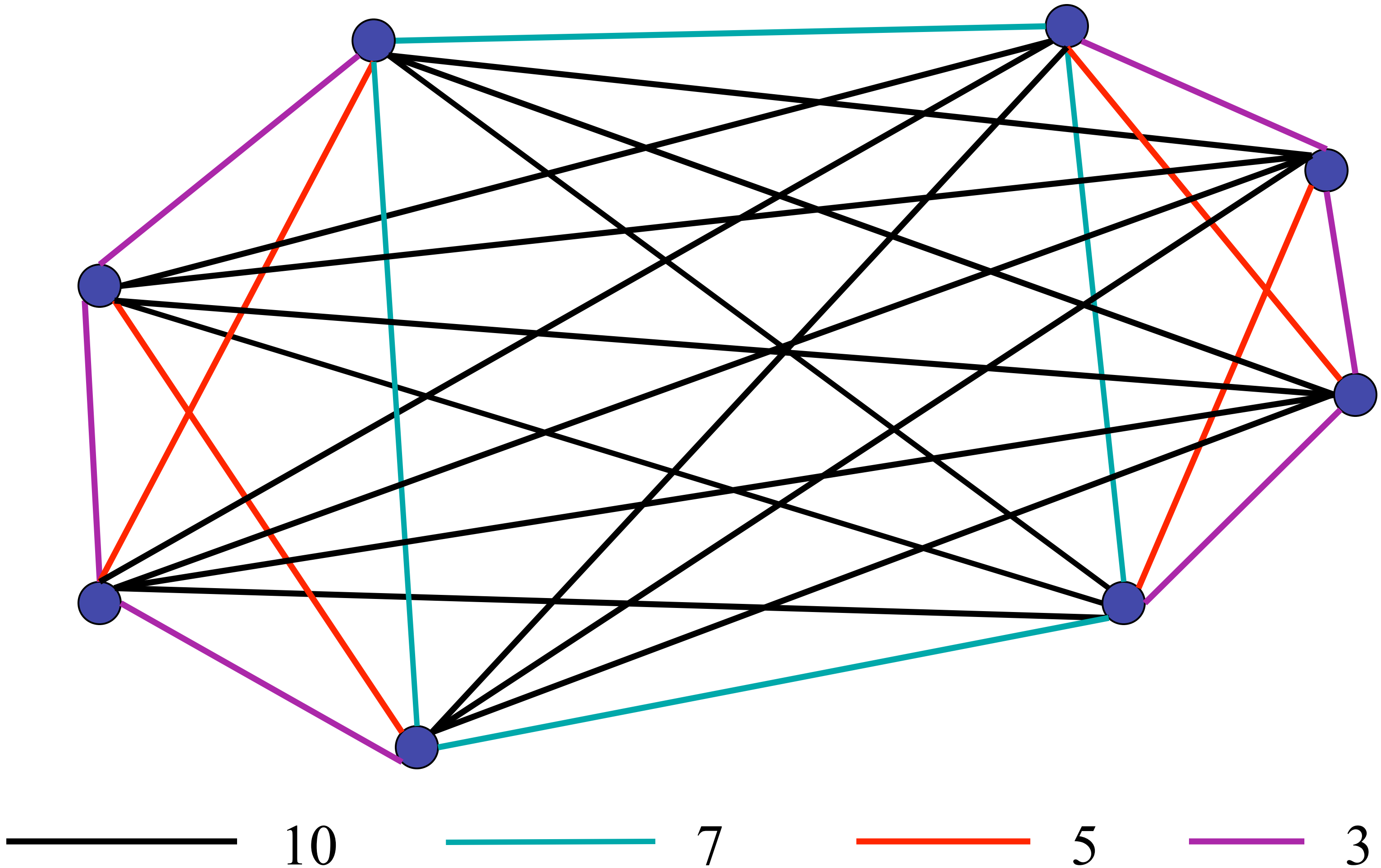
Комбинаторные алгоритмы

Параметрическое сокращение

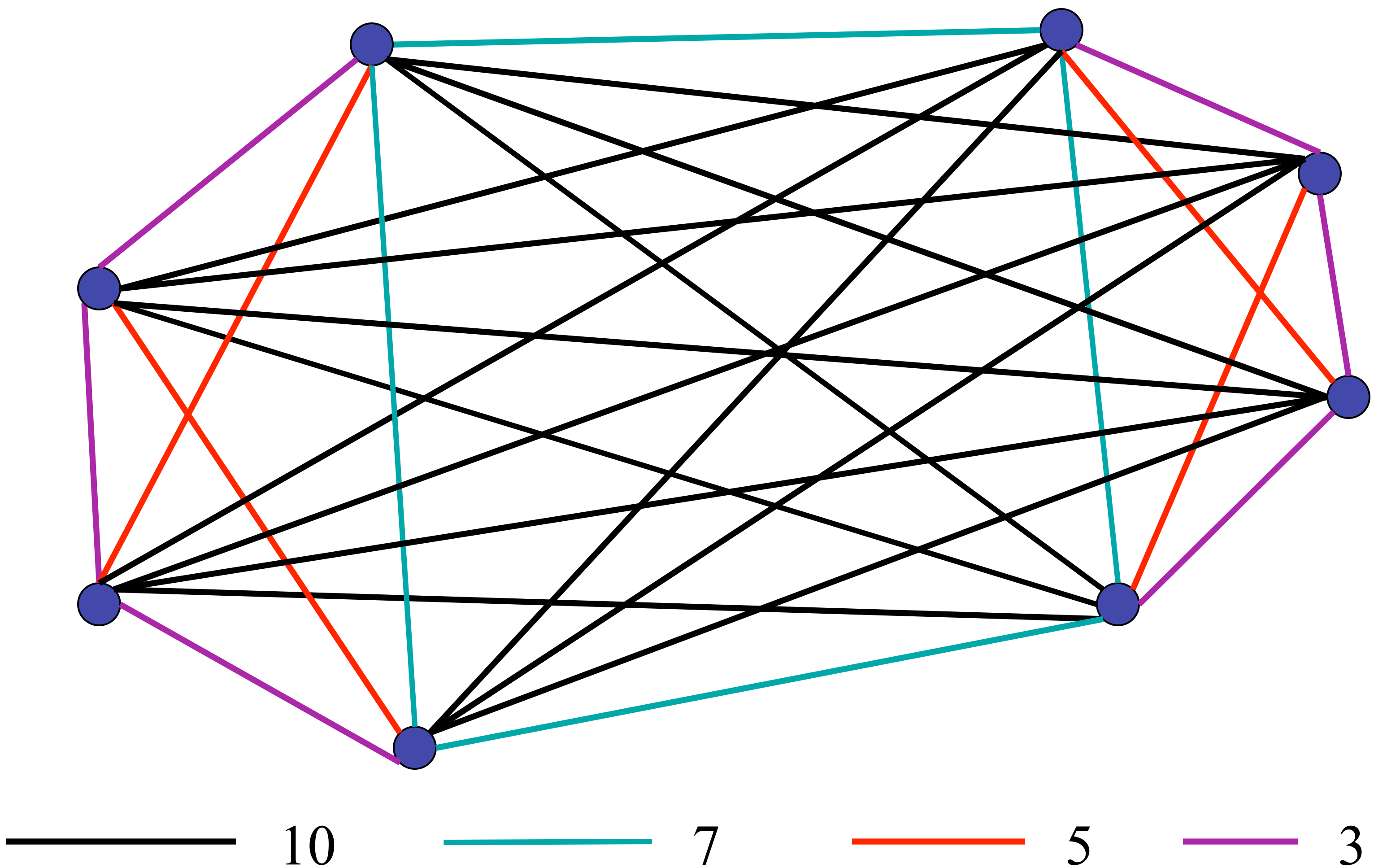
Метрическая задача о k центрах

- *Дано*: Полный граф $G = (V, E)$, стоимости ребер $cost: E \rightarrow \mathbf{Q}^+$ такие, что для любых трех вершин u, v и w: $cost(u,v) \leq cost(u,w) + cost(w,v)$. Для любого множества $S \subseteq V$, определим $connect(v,S) = \min\{cost(u,v) | u \in S\}$.
- *Найти* множество $S \subseteq V$, с $|S|=k$, которое минимизирует величину $\max_v\{connect(v,S)\}$.
- Метрическая задача о k центрах – NP-трудна.

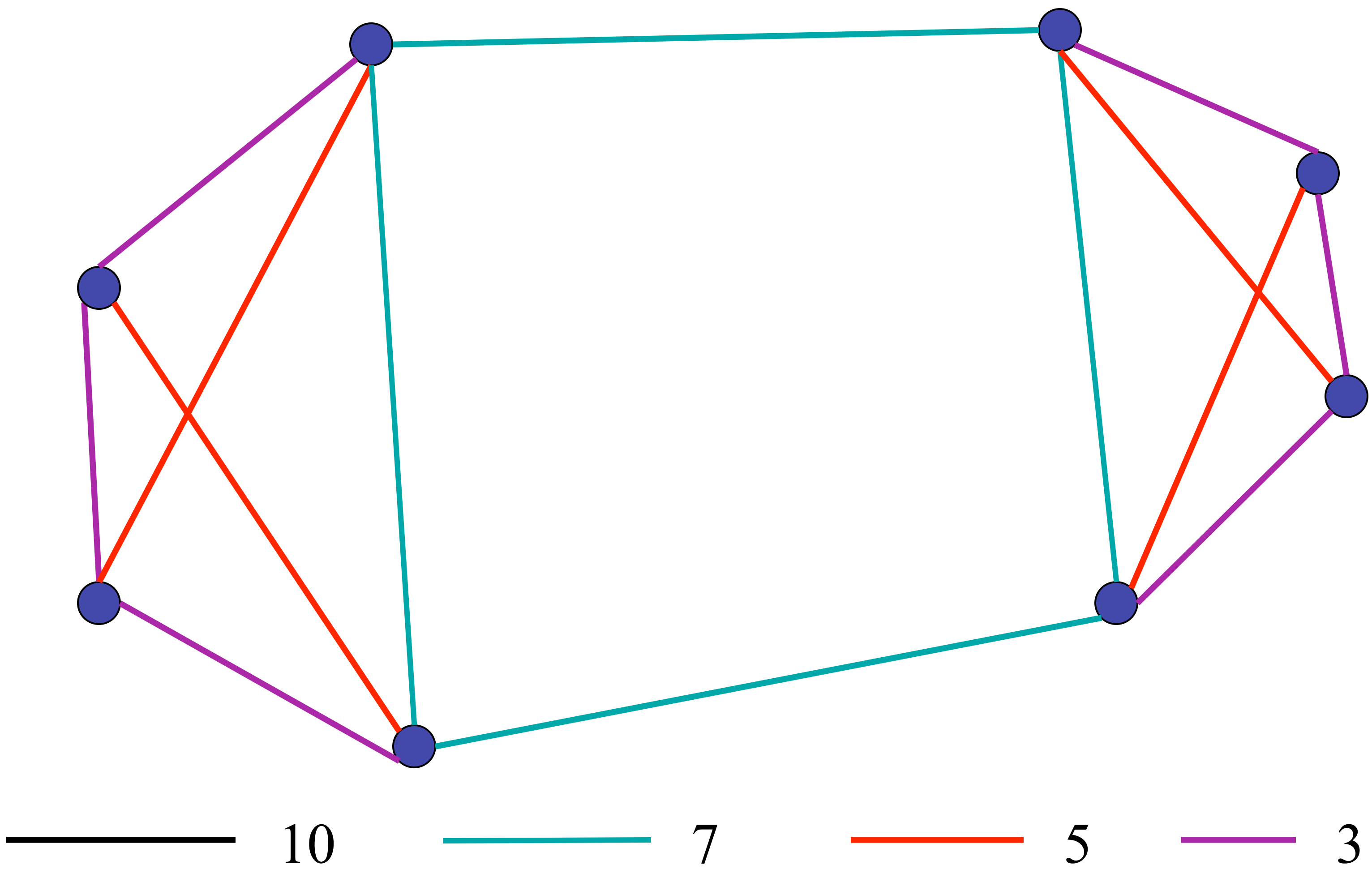
Пример



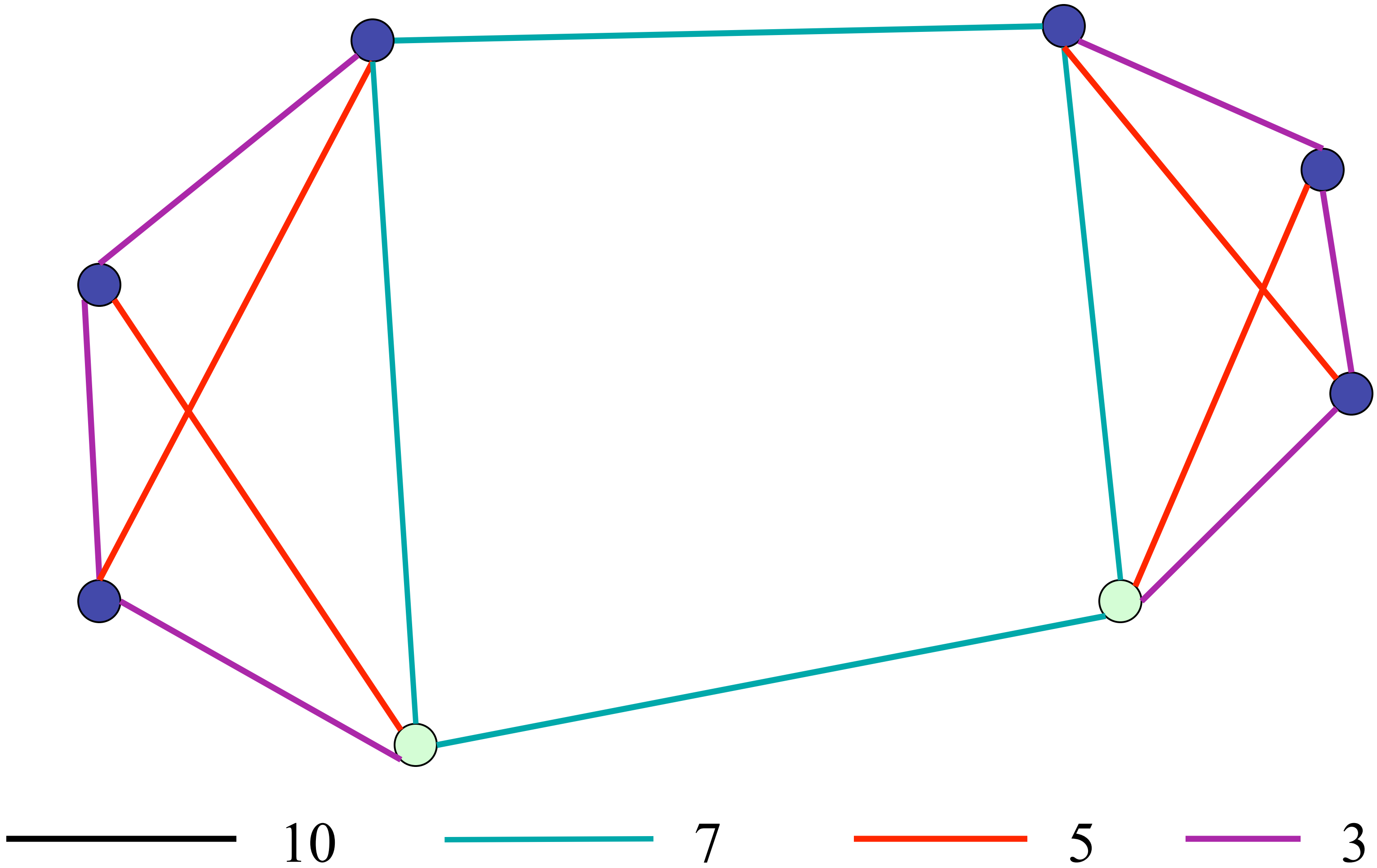
Идея алгоритма (k=2, OPT ≤ 7)



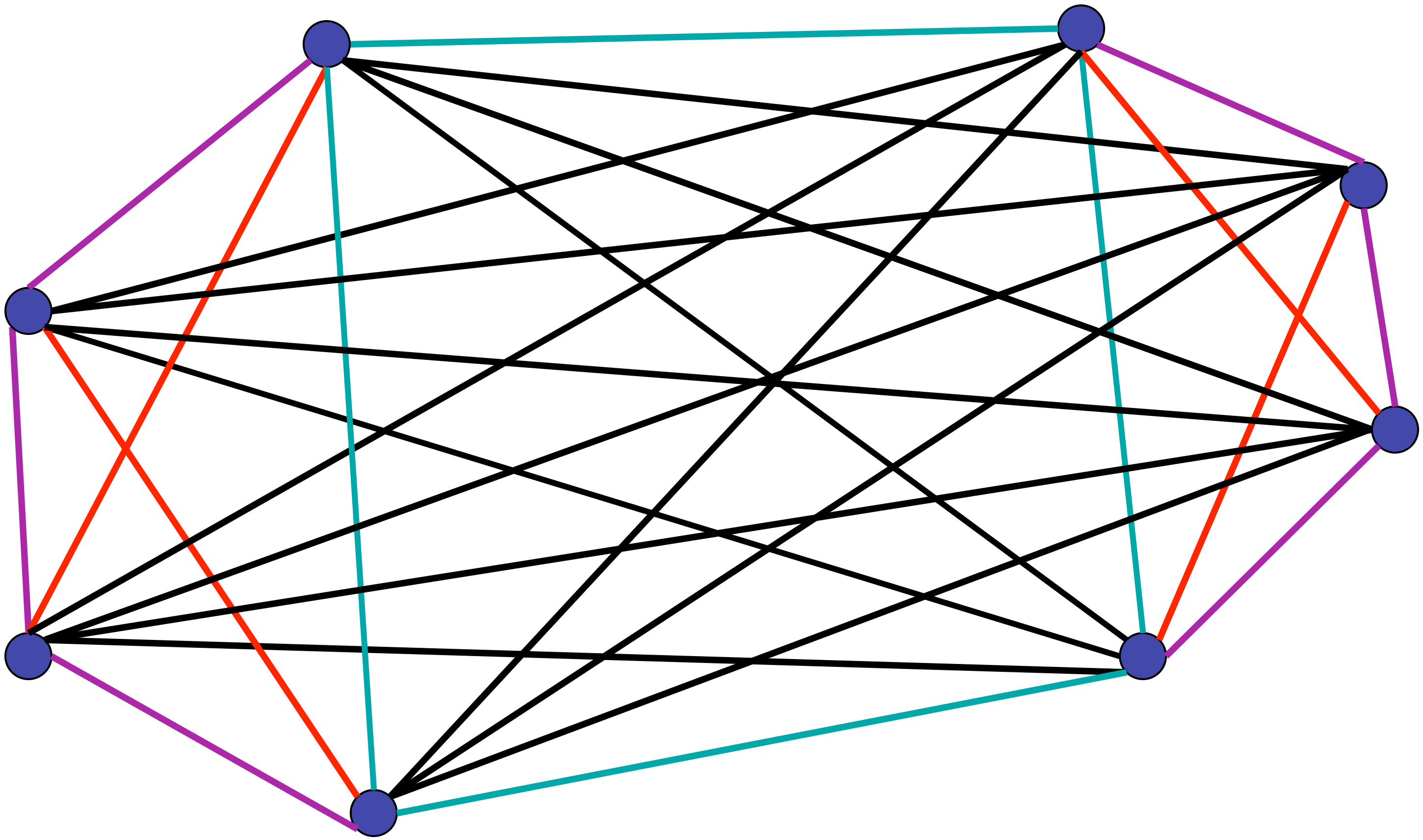
Idea of Algorithm (k=2, OPT ≤ 7)



Idea of Algorithm (k=2, OPT ≤ 7)

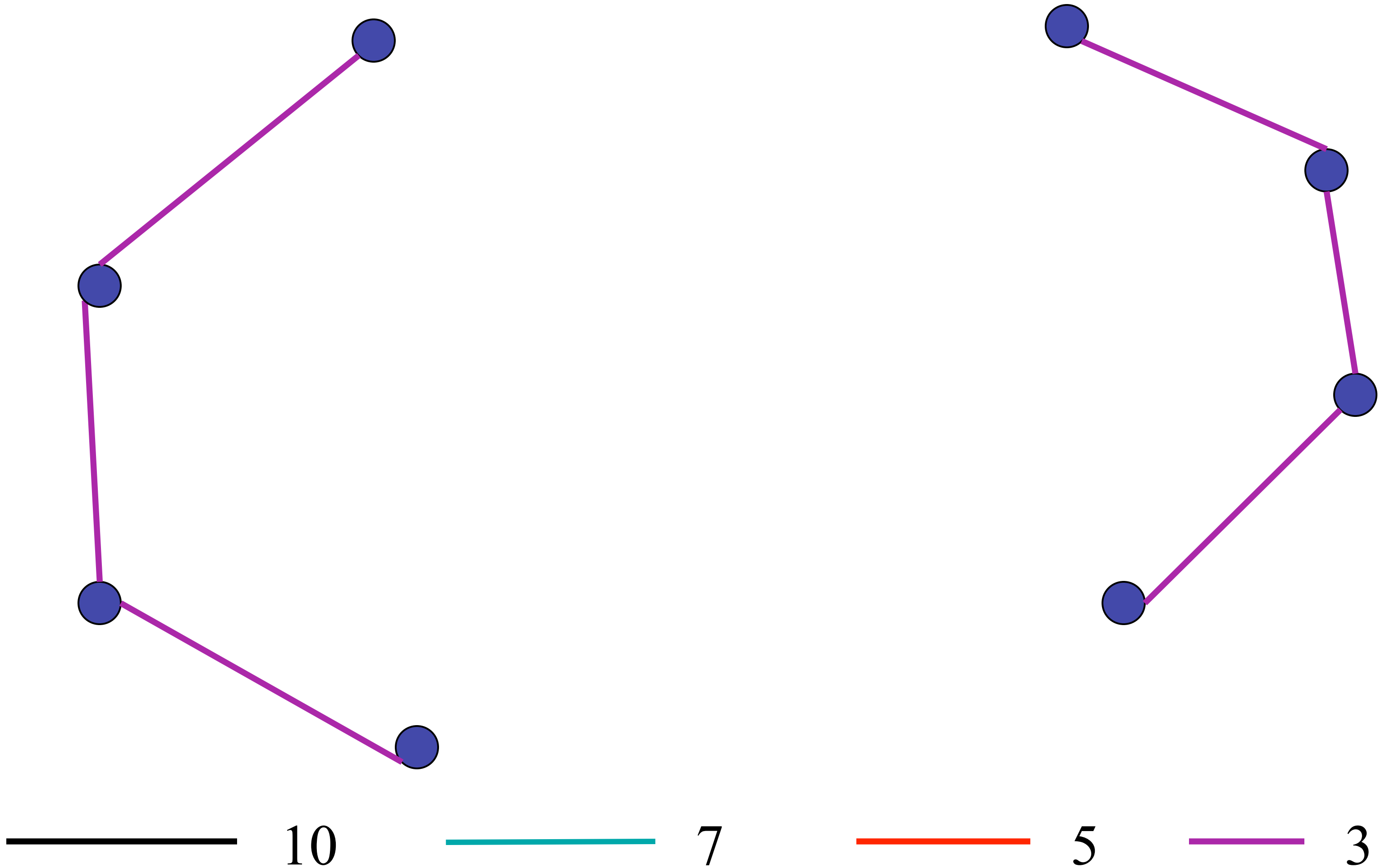


Идея алгоритма (k=2, OPT ≤ 3)



10 — 7 — 5 — 3

Idea of Algorithm (k=2, OPT ≤ 3)



Параметрическое сокращение

- Упорядочим ребра в G по невозрастанию их стоимости $cost(e_1) \leq cost(e_2) \leq \ldots \leq cost(e_m)$.
- Пусть $G_i = (V, E_i)$, где $E_i = \{e_1, e_2, \ldots, e_i\}$.
- Для каждого G_i, нужно проверить существует ли множество $S \subseteq V$, с которым смежна каждая вершина из $V - S$.

Доминирующее множество

- **Доминирующим множеством** в графе $G = (V, E)$ называется подмножество вершин $S \subseteq V$ такое, что каждая вершина в $V - S$ смежна некоторой вершине в S.
- $\mathrm{dom}(G)$ – размер доминирующего множества минимальной мощности.

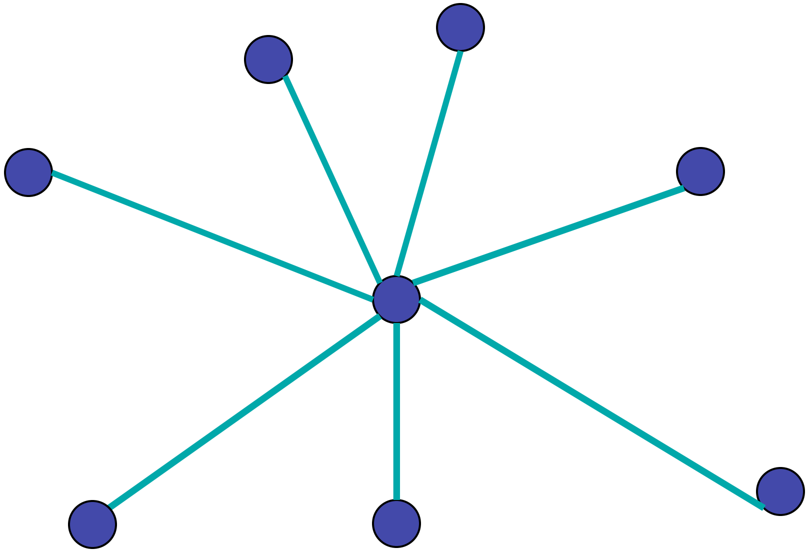
Доминирующее множество

- **Доминирующим множеством** в графе $G = (V, E)$ называется подмножество вершин $S \subseteq V$ такое, что каждая вершина в $V - S$ смежна некоторой вершине в S.
- dom(G) – размер доминирующего множества минимальной мощности.
- Вычисление dom(G) – NP-трудная задача.

Задача о k центрах

- Задача о k центрах эквивалентна задаче нахождения наименьшего индекса i такого, что G_i имеет доминирующее множество размера k.
- G_i содержит k звезд ($K_{1,p}$), которые охватывают все вершины.

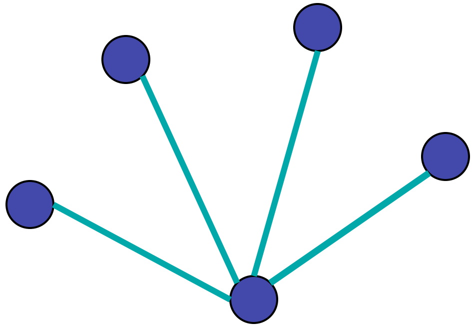
$K_{1,7}$



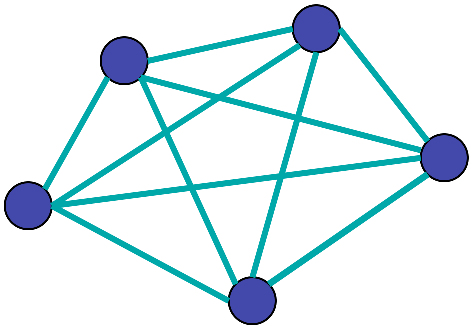
G^2

- **Независимым множеством** в графе $G = (V, E)$ называется подмножество вершин $I \subseteq V$ такое, что в нем нет смежных вершин.
- **Квадратом** графа $G = (V, E)$ называется граф $G^2 = (V, E')$, где $(u,v) \in E'$, когда длина пути между вершинами u и v меньше или равна 2.

$\boldsymbol{G}=K_{1,4}$



$G^2=K_5$



Нижняя оценка

- **Лемма 4.1**

  Дан граф H, пусть I будет независимое множество в H^2. Тогда $| I | \leq \mathrm{dom}(H)$.

Алгоритм Хошбаум-Шмойса

Input (G, $cost$: $E \rightarrow \mathbf{Q}^+$)

1) Строим $G_1^2, G_2^2, \ldots, G_m^2$.
2) Найти максимальное независимое множество I_r в каждом графе G_r^2.
3) Вычислить наименьший индекс r такой, что $|\, I_r \,| \leq k$. Пусть это будет j.

Output (I_j)

Оценка качества алгоритма Хошбаум-Шмойса

Теорема 4.2

Алгоритм Хошбаум-Шмойса является 2-приближенным алгоритмом для метрической задачи о k центрах.

Ключевая лемма

- **Лемма 4.3**

  Для j, определенного алгоритмом, $cost(e_j) \le \mathrm{OPT}$.

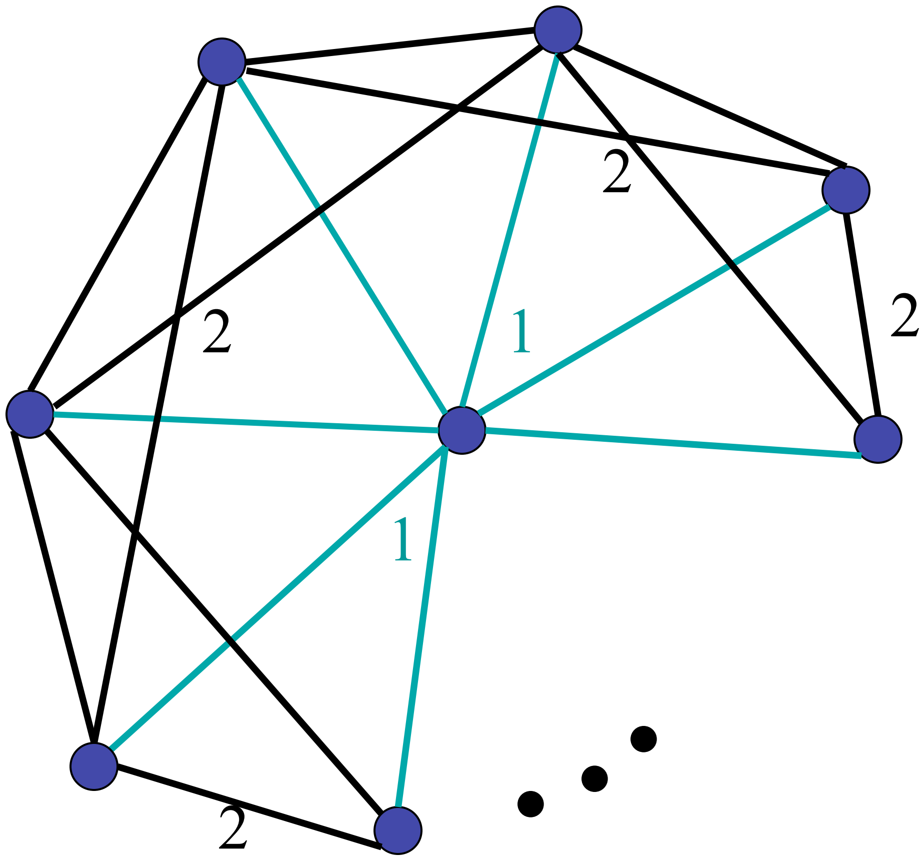
Доказательство.

- Для каждого $r < j$ имеем $|I_r| > k$.
- Тогда по лемме 6.1 $\mathrm{dom}(G_r) \ge |I_r| > k$.
- Поэтому $r^* > r$, и $r^* \ge j$.
- $cost(e_j) \le \mathrm{OPT}$

Доказательство Теоремы 4.2

- Максимальное независимое множество является также и доминирующим.
- В графе G_j^2 найдутся звезды с центрами в вершинах I_j, которые покрывают все вершины G.
- Из неравенства треугольника стоимость ребер в G_j^2 не превосходит $2cost(e_j)$.
- По лемме 6.3: $2\ cost(e_j) \leq 2$ OPT.

Точность оценки



Метрическая задача о взвешенных центрах

- *Дано*: Полный граф $G = (V, E)$, стоимости ребер $cost: E \to \boldsymbol{Q}^+$ такие, что для любых трех вершин u, v и w: $cost(u,v) \leq cost(u,w) + cost(w,v)$ и $w: V \to \boldsymbol{R}^+$, и граница $W \in \boldsymbol{R}^+$.
- *Найти* множество $S \subseteq V$, суммарного веса не больше W, которое минимизирует величину $\max_v\{connect(v,S)\}$.

Взвешенное доминирующее множество

- **Доминирующим множеством** в графе $G = (V, E)$ называется подмножество вершин $S \subseteq V$ такое, что каждая вершина в $V - S$ смежна некоторой вершине в S.
- wdom(G) – вес доминирующего множества минимального веса в G.
- Вычисление wdom(G) – NP-трудная задача.

Параметрическое сокращение

- Упорядочим ребра в G по невозрастанию их стоимости $cost(e_1) \leq cost(e_2) \leq \ldots \leq cost(e_m)$.
- Пусть $G_i = (V, E_i)$, где $E_i = \{e_1, e_2, \ldots, e_i\}$.
- Требуется найти наименьший индекс i такой, что G_i имеет доминирующее множество веса не больше W, то есть $\mathrm{wdom}(G_i) \leq W$.

Лёгкие соседи

- **Расстоянием** ($\mathrm{dist}(v,u)$, $\mathrm{dist}_G(v,u)$) для двух вершин v и u называется длина кратчайшего v-u-пути в G.
- $\mathrm{Neighbor}_G(u) = \{v \mid \mathrm{dist}_G(u,v) \le 1\}$
- Дан граф $G = (V, E)$, и w: $V \to \boldsymbol{R}^+$, пусть I будет независимое множество в G^2.
- Пусть $s(u) \in \mathrm{Neighbor}_G(u)$ обозначает соседа u в G наименьшего веса.
- $S = \{s(u) \mid u \in I\}$

Нижняя оценка

- **Лемма 4.4**

Дан граф H, пусть I будет независимое множество в H^2. Тогда $w(S) \le \mathrm{wdom}(H)$.

Доказательство.

- Пусть D доминирующее множество минимального веса в H.
- Тогда $\exists$ набор непересекающихся звезд в H с центрами в D и покрывающими все вершины.
- Так как, каждая звезда превращается в клику в H^2, то в I попадет не более одной вершины из такой клики.
- Каждая попавшая вершина является в исходном графе соседом центра из $D \Rightarrow w(S) \le \mathrm{wdom}(H)$.

Алгоритм Хошбаум-Шмойса-2

Input (G, $cost$: $E \to \mathbf{Q}^+$, w: $V \to \boldsymbol{R}^+$, W)

1) Построить $G_1^{\,2}, G_2^{\,2},\ldots, G_m^{\,2}$.
2) Найти максимальное независимое множество I_r в каждом графе $G_r^{\,2}$.
3) Построить $S_r = \{s_r(u) \mid u \in I_r\}$
4) Вычислить наименьший индекс r такой, что $w(S_r) \le W$. Пусть это будет j.

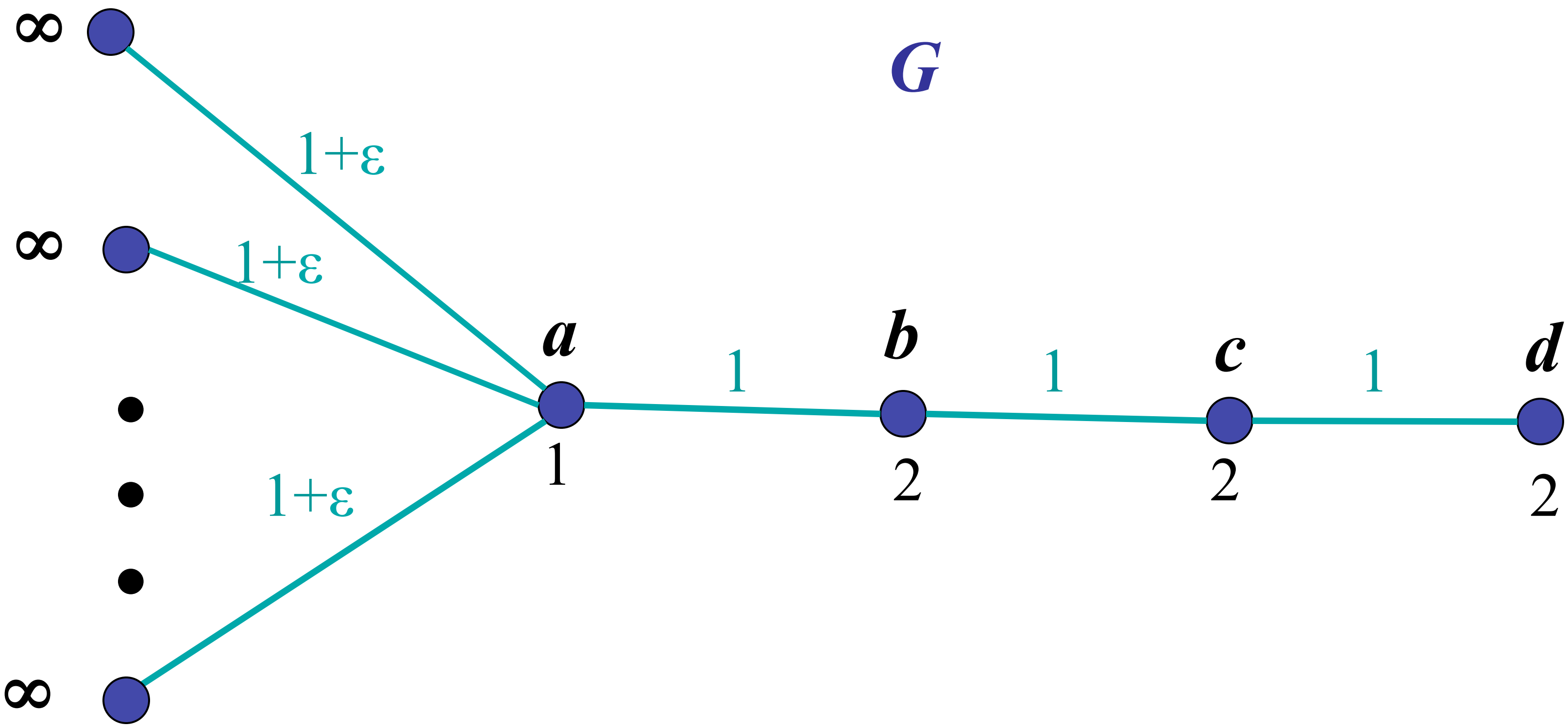
Output (S_j)

Оценка качества алгоритма Хошбаум-Шмойса-2

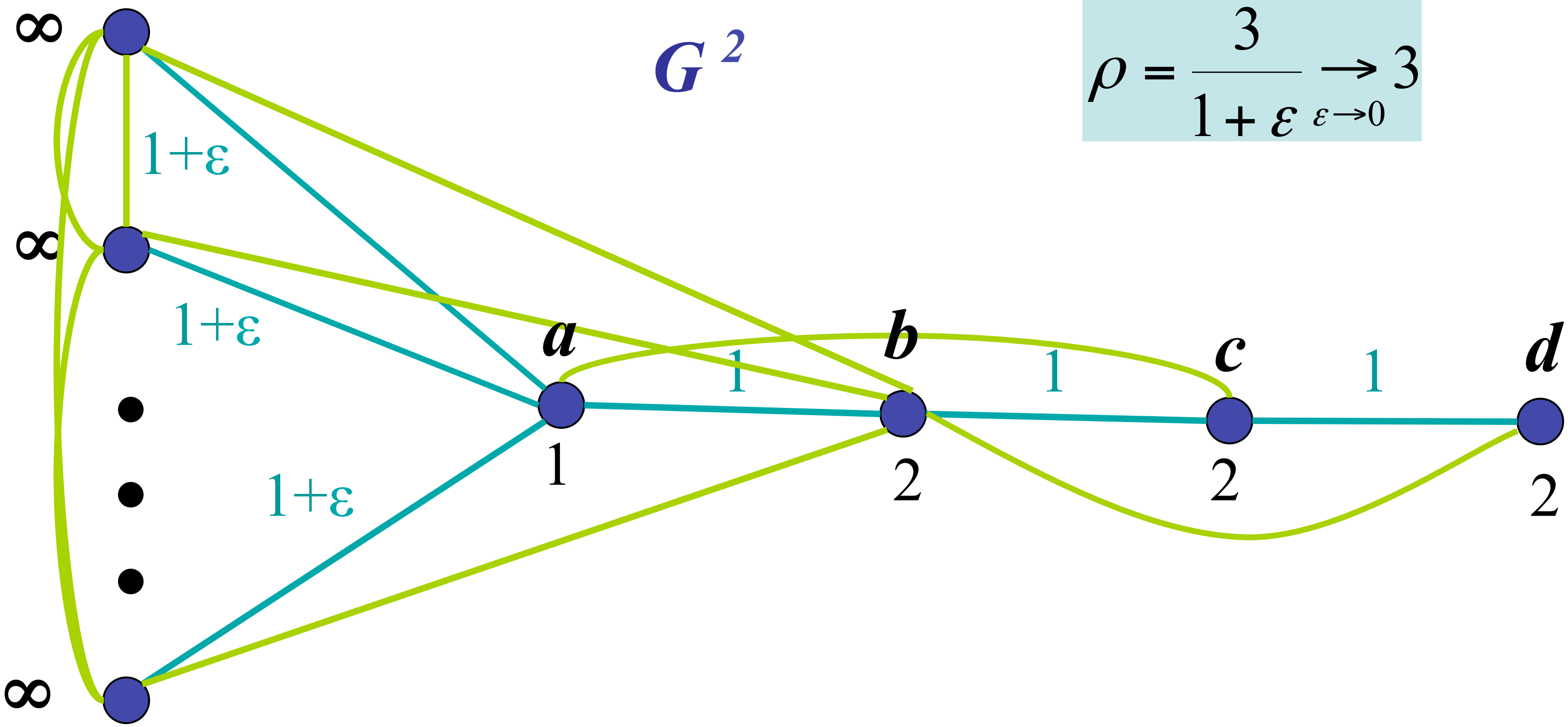
Теорема 4.5

Алгоритм Хошбаум-Шмойса является 3-приближенным алгоритмом для метрической задачи о k взвешенных центрах.

Точность оценки



Точность оценки



$$\rho = \frac{3}{1+\varepsilon} \xrightarrow[\varepsilon \to 0]{} 3$$

$I_{n+3}=\{b\}$

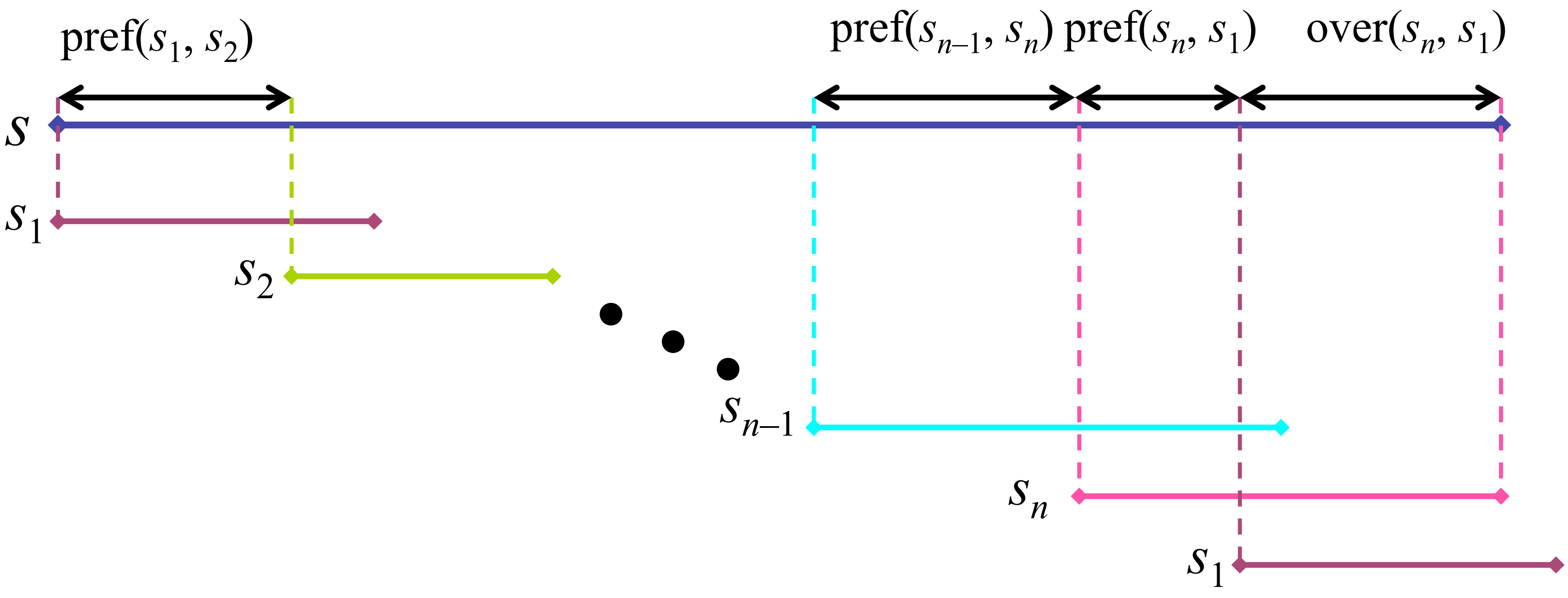
$S_{n+3}=\{a\}$

$OPT=\{a, c\}$

Задача «Кратчайшая суперстрока»

- *Дано*: Конечный алфавит Σ и множество из n строк $S = \{s_1, \ldots, s_n\} \subseteq \Sigma^+$.
- *Найти* кратчайшую суперстроку s, которая содержит каждую строку s_i, как подстроку.
- Без ограничения общности будем считать, что никакая строка s_i не содержит другую строку s_j, $i \neq j$, как подстроку.

Сумма префиксов



$$\mathrm{OPT} = |\mathrm{prefix}(s_1, s_2)| + |\mathrm{prefix}(s_2, s_3)| + \cdots +$$
$$+ |\mathrm{prefix}(s_n, s_1)| + |\mathrm{overlap}(s_n, s_1)|.$$

Ориентированный граф префиксов

- $G_{pref} = (V, A)$ – полный орграф.
- $V=\{s_1, \ldots, s_n\}=\{1,\ldots,n\}$
- дуга (s_i, s_j) имеет вес $prefix(s_i,s_j)$.
- $|prefix(s_1,s_2)| + |prefix(s_2,s_3)| + \ldots + |prefix(s_n,s_1)|$ – вес тура $s_1 \to s_2 \to \ldots \to s_n \to s_1$ $(1 \to 2 \to \ldots \to n \to 1)$.
- Минимальный по весу гамильтонов цикл – нижняя оценка на длину кратчайшей суперстроки s.

Нижняя оценка

- **Циклическое покрытие** – набор непересекающихся циклов, покрывающих все вершины.
- Гамильтонов цикл также является циклическим покрытием.
- Циклическое покрытие минимального веса – нижняя оценка на длину кратчайшей суперстроки.
- Циклическое покрытие минимального веса можно вычислить за полиномиальное время.

От цикла к префиксам

- Если $c = (i_1 \rightarrow i_2 \rightarrow \ldots \rightarrow i_l \rightarrow i_1)$ – цикл в префиксном графе, то $\alpha(c) = prefix(s_{i_1}, s_{i_2}) \circ \ldots \circ prefix(s_{i_{l-1}}, s_{i_l}) \circ prefix(s_{i_l}, s_{i_1})$.
- $|\alpha(c)|$ – вес цикла c.
- Каждая из строк $s_{i_1}, s_{i_2}, \ldots, s_{i_l}$ – это подстрока в $(\alpha(c))^\infty$.
- $\sigma(c) = \alpha(c) \circ s_{i_1}$.
- $\sigma(c)$ – суперстрока над $s_{i_1}, s_{i_2}, \ldots, s_{i_l}$.
- s_{i_1} – первая строка в цикле c.

Пример

abcdeabcdeabcde
bcdeabcdeabcdea
cdeabcdeabcdeabc
deabcdeabcdeabcd
abcdeabcdeabcde

$\alpha(c)$=abcde , $(\alpha(c))^2$=abcdeabcde ,
bcdeabcdeabcdea – подстрока $(\alpha(c))^4$.

$\sigma(c)$ = abcdeabcdeabcdeabcde

Алгоритм «Суперстрока»

Input $(s_1, \ldots, s_n)$

1) Построить префиксный граф G_{pref} для $s_1, \ldots, s_n$.
2) Найти минимальное по весу циклическое покрытие G_{pref}, $C = \{c_1, \ldots, c_k\}$

Output $(\sigma(c_1) \circ \ldots \circ \sigma(c_k))$.

Замечание

- Очевидно, что $\sigma(c_1)\circ\ldots\circ\sigma(c_k)$ является суперстрокой.
- Более того, если в каждом цикле есть строка, длина которой не превосходит вес цикла, то решение не превосходит 2OPT.
- Таким образом, решение может быть плохим, если все строки в цикле длинные.

Оценка веса покрывающего цикла

- **Лемма 4.6**

  Если каждая строка в $S' \subseteq S$ является подстрочкой t^∞ для строки t, то существует цикл веса не больше $|t|$ в префиксном графе, покрывающий все вершины, соответствующие строкам в S' .

Доказательство леммы 4.6

- Упорядочим строки из S' по моментам их первого появления в t^∞ .
- Все эти моменты различны и появляются в первой копии t.
- Рассмотрим цикл в префиксном графе, посещающий вершины в заданном порядке.
- Ясно, что его вес не больше чем t.

Оценка мощности пересечения двух первых строк

- **Лемма 4.7**

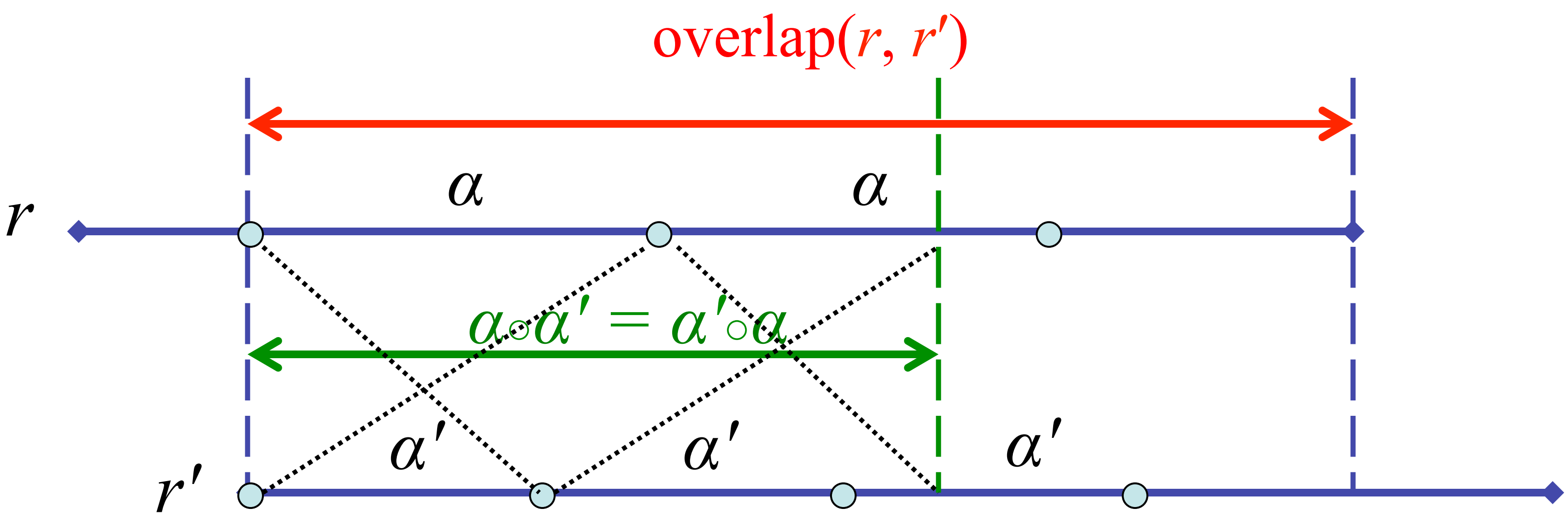
Пусть c и c' два цикла в C (циклическое покрытие минимального веса), и пусть r, r' первые строки этих циклов. Тогда

$$|\mathrm{overlap}(r, r')| < w(c) + w(c').$$

$$|\mathrm{overlap}(r, r')| \geq w(c) + w(c').$$

α – префикс длины $w(c)$ в пересечении r и r'.

α' – префикс длины $w(c')$ в пересечении r и r'.



$(\alpha)^\infty = (\alpha')^\infty$

$\exists$ цикл длины $w(c)$ в префиксном графе, покрывающем все строки в c и c'.

Оценка качества алгоритма «Суперстрока»

Теорема 4.8

Алгоритм «Суперстрока» является 4-приближенным алгоритмом для задачи «Кратчайшая суперстрока».

Алгоритм «Суперстрока»

Input $(s_1, \ldots, s_n)$

1) Построить префиксный граф G_{pref} для $s_1, \ldots, s_n$.
2) Найти минимальное по весу циклическое покрытие G_{pref}, $C = \{c_1, \ldots, c_k\}$

Output $(\sigma(c_1) \circ \ldots \circ \sigma(c_k))$.

Доказательство

$$w(C) = \sum_{i=1}^{k} w(c_i) \le \mathrm{OPT}$$

$$A = \sum_{i=1}^{k} \sigma(c_i) = w(C) + \sum_{i=1}^{k} |r_i|$$

r_i – первая строка в c_i.

$$string^* : \ldots r_1, \ldots, r_2, \ldots, r_k, \ldots$$

$$\mathrm{OPT} \ge \sum_{i=1}^{k} |r_i| - \sum_{i=1}^{k-1} |\mathrm{overlap}(r_i, r_{i+1})| \ge$$

$$\stackrel{L\,6.4}{\ge} \sum_{i=1}^{k} |r_i| - 2\sum_{i=1}^{k} w(c_i)$$

$$\sum_{i=1}^{k} |r_i| \le 3\,\mathrm{OPT}$$

$$A \le 4\,\mathrm{OPT}$$

Упражнение 4.1

- Предложите 2-приближенный алгоритм для следующей задачи.
- *Дано*: Полный граф $G = (V, E)$, стоимости ребер $cost: E \to \mathbf{Q}^+$ такие, что для любых трех вершин u, v и w: $cost(u,v) \le cost(u,w) + cost(w,v)$.
- *Найти* разбиение множества вершин V на k подмножеств $V_1, \ldots, V_k$ такое, что $\max\{cost(u,v) | 1 \le i \le k,\ u,v \in V_i\}$ минимален.