Комбинаторные алгоритмы

Задача о покрытии

Задача о покрытии множествами

- *Дано*: Совокупность U из n элементов, и набор подмножеств U, $S = \{S_1,\ldots, S_k\}$, и веса(стоимости) подмножеств c: $S \rightarrow \mathbf{Q}^+$.
- *Найти* покрытие наименьшего веса.
- $S' \subseteq S$ является покрытием, если любой элемент из U принадлежит хотя бы одному элементу из S' .

Кратность

- Пусть f_i – кратность элемента e_i, то есть число множеств из S, в которые он входит.
- Пусть $f = \max_{i=1,\ldots,n} f_i$.

Жадная стратегия

- Пусть C будет множество элементов уже покрытое на предыдущих итерациях. Тогда $\alpha_i = c(S_i)/|S_i - C|$ называется **эффективностью** множества S_i и равна **средней стоимости**, с которой покрывается новый элемент.
- Назовем **ценой** элемента e среднюю стоимость с которой он был покрыт.
- Эквивалентно, когда множество S выбрано, мы можем понимать, что его стоимость равномерно распределена среди вновь покрытых элементов.

Алгоритм Хватала

0) **Input** (U, S, c: $S \to \mathbf{Q}^+$)

1) $C \leftarrow \varnothing$, $Sol \leftarrow \varnothing$

2) **While** $C \neq U$ **do:**

   Найти $S_i \in S - Sol$, у которого

   $\alpha_i = c(S_i)/|S_i - C|$ минимально.

   $Sol \leftarrow Sol \cup \{S_i\}$

   $C \leftarrow C \cup S_i$ (S_i – самое эффективное)

   $price(e) = \alpha_i$ для всех $e \in S_i - C$

3) **Output** (Sol)

Анализ алгоритма Хватала

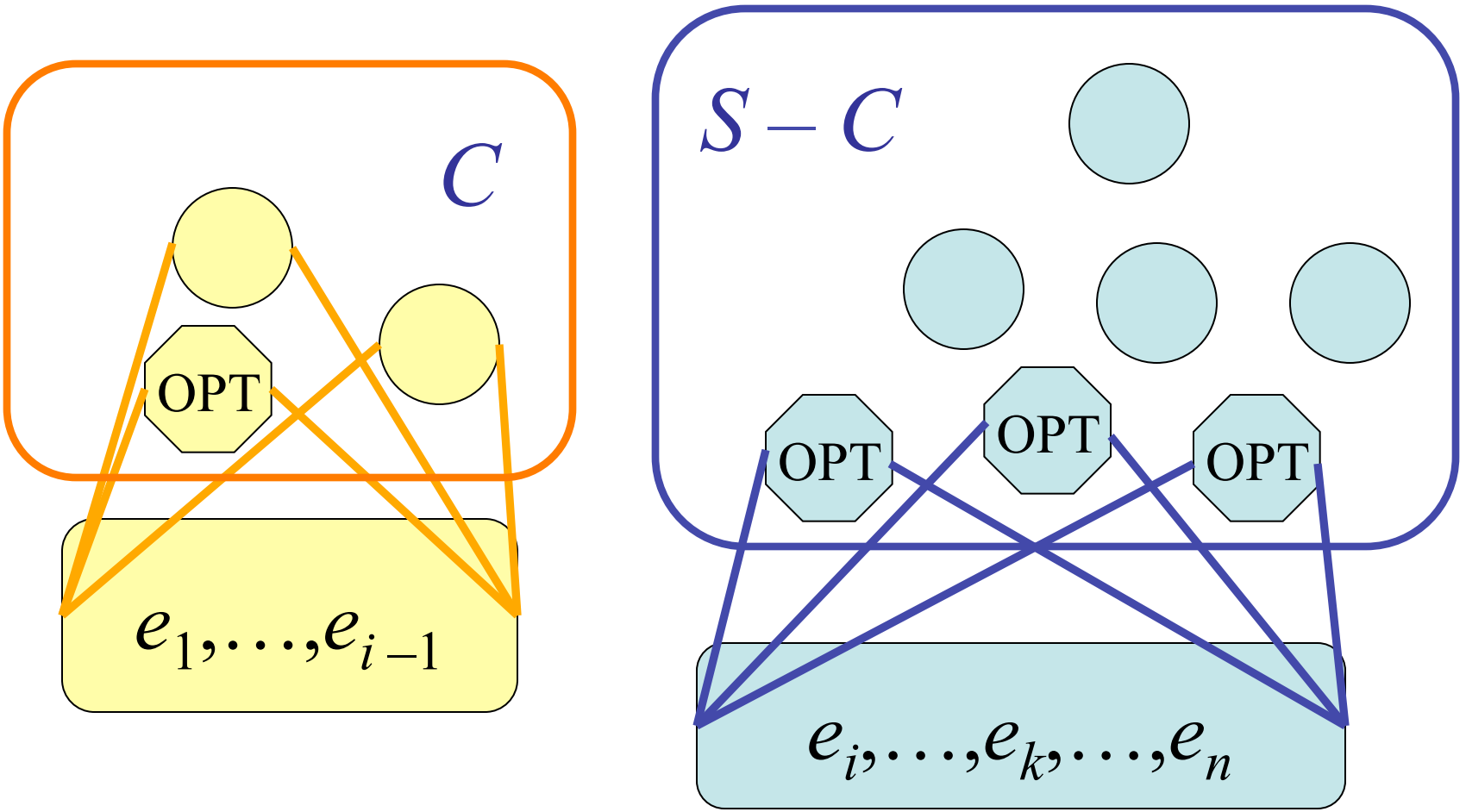
- Упорядочим элементы из U в порядке, в котором они были покрыты алгоритмом.
- Пусть это будет $e_1,\ldots,e_n$.

- **Лемма 2.1**

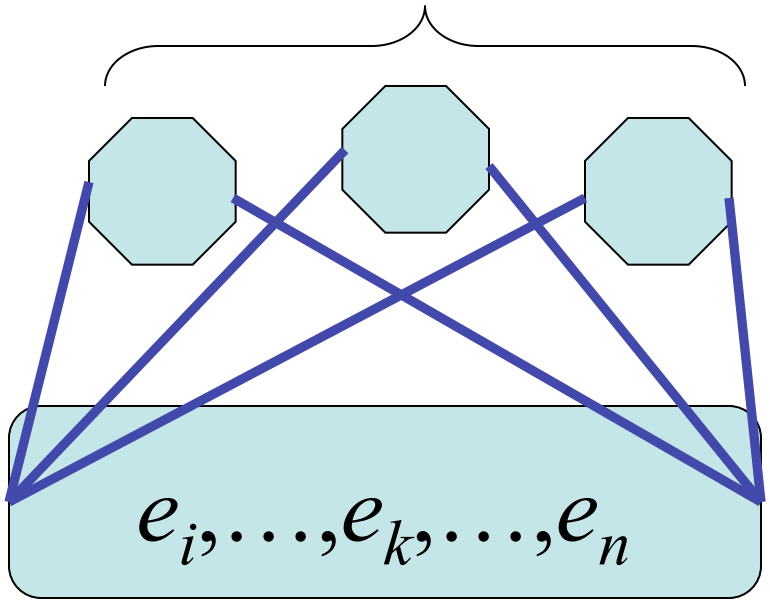
  Для каждого $k \in \{1,\ldots,n\}$,

  $price(e_k) \leq \mathrm{OPT}/(n-k+1)$.

Доказательство леммы 2.1

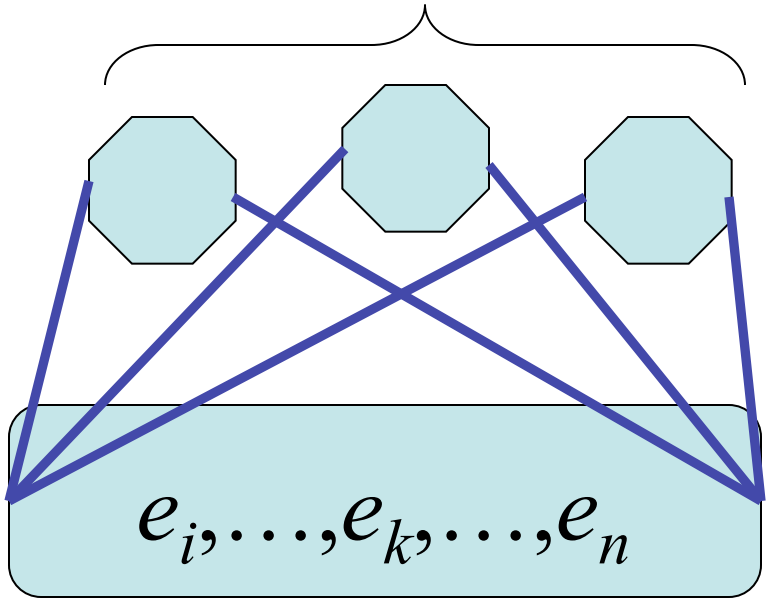


Доказательство леммы 2.1



Доказательство леммы 2.1

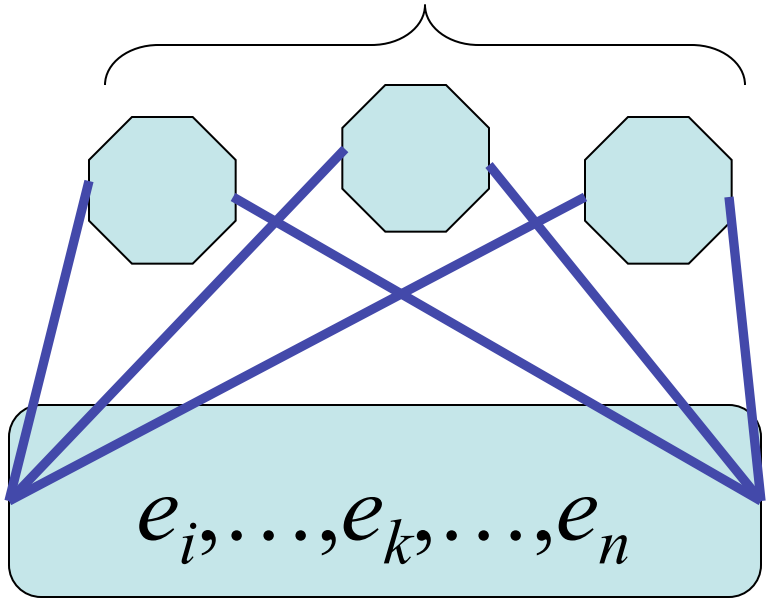
Общая эффективность не больше чем OPT/($n - i + 1$) ≤ OPT/($n - k + 1$)



Доказательство леммы 2.1

$$\frac{a_1}{b_1} \geq \frac{a_2}{b_2} \geq \cdots \geq \frac{a_n}{b_n} \Rightarrow \frac{a_1 + \cdots + a_n}{b_1 + \cdots + b_n} \geq \frac{a_n}{b_n}$$

Общая эффективность не больше чем OPT/($n - i + 1$) ≤ OPT/($n - k + 1$)



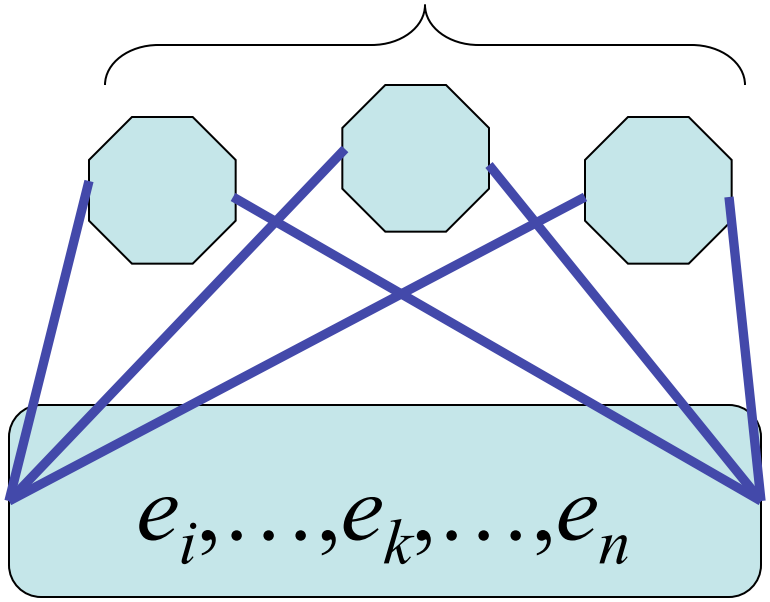
Существует $S_j \in S - C$ с $\alpha_j \leq$ OPT/($n - k + 1$).

Доказательство леммы 2.1

$$\frac{a_1}{b_1} \geq \frac{a_2}{b_2} \geq \cdots \geq \frac{a_n}{b_n} \Rightarrow \frac{a_1 + \cdots + a_n}{b_1 + \cdots + b_n} \geq \frac{a_n}{b_n}$$

Общая эффективность не больше чем OPT/$(n - i + 1)$ ≤ OPT/$(n - k + 1)$

Существует $S_j \in S - C$ с $\alpha_j \leq$ OPT/$(n - k + 1)$.



$price(e_k) \leq$ OPT/$(n-k+1)$.

Оценка качества алгоритма Хватала

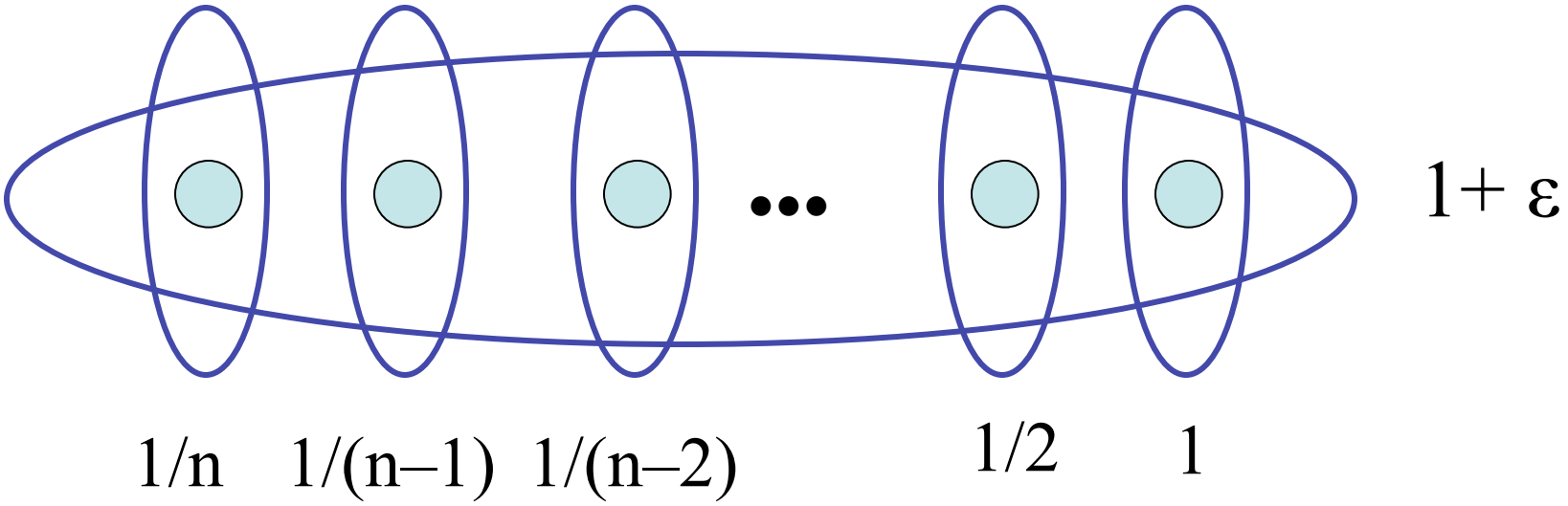
Теорема 2.2

Алгоритм Хватала является H_n-приближенным алгоритмом для задачи о покрытии множествами, где $H_n=1+1/2+1/3+\ldots+1/n$.

Доказательство:

$$\sum_{S_i \in C} c(S_i) = \sum_{k=1}^{n} price(e_k) \leq \left(1 + \frac{1}{2} + \cdots + \frac{1}{n}\right) OPT$$

Точность оценки



Задача о вершинном покрытии

- *Дано*: Граф $G = (V, E)$, веса вершин w: $V \to \mathbf{Q}^+$.
- *Найти* вершинное покрытие наименьшего веса.

Пропорционально-степенная функция

- Назовем функцию, соответствующую весам вершин, **пропорционально-степенной**, если существует константа $c > 0$, такая что вес каждой вершины $v \in V$ равен $c \cdot \deg(v)$.

Нижняя оценка

- **Лемма 2.3**

  Пусть w: $V \to \mathbf{Q}^+$ пропорционально-степенная функция. Тогда $w(V) \leq 2$ OPT.

Доказательство

- c: $w(c) = c \cdot \deg(v)$,
- U оптимальное вершинное покрытие G.

$$\sum_{v \in U} \deg(v) \geq |E|.$$

- Тогда $w(U) \geq c|E|$.

$$\sum_{v \in V} \deg(v) = 2|E|, \quad w(V) = 2c|E|.$$

Наибольшая пропорционально-степенная функция

- Пусть w: $V \to \mathbf{Q}^+$ произвольная функция.
- Определим **наибольшую пропорционально-степенную функцию относительно *w*** в графе G следующим образом:
  - Удалим из графа все вершины степени 0. Среди оставшихся вершин вычислим $c = \min\{w(v)/\deg(v)\}$.
  - Тогда $t(v) = c \cdot \deg(v)$ есть искомая функция
- Определим через $w'(v) = w(v) - t(v)$ **остаточную функцию весов**.

Алгоритм «Уровневый»

0) **Input** ($G = (V, E)$, $w: V \to \mathbf{Q}^+$)

1) $Sol \leftarrow \varnothing$, $i \leftarrow 0$, $w'(v) \leftarrow w(v)$,
$V_0 \leftarrow V - D_0$ ($D_0 = \{v \in V \mid \deg(v)=0\}$)

2) **While** $V_i \neq \varnothing$ **do:**

$c \leftarrow \min\{w(v)/\deg(v)\}$

$t(v) \leftarrow c \cdot \deg(v)$

$w'(v) \leftarrow w'(v) - t(v)$

$Sol \leftarrow Sol \cup W_i$ ($W_i = \{v \in V_i \mid w'(v)=0\}$)

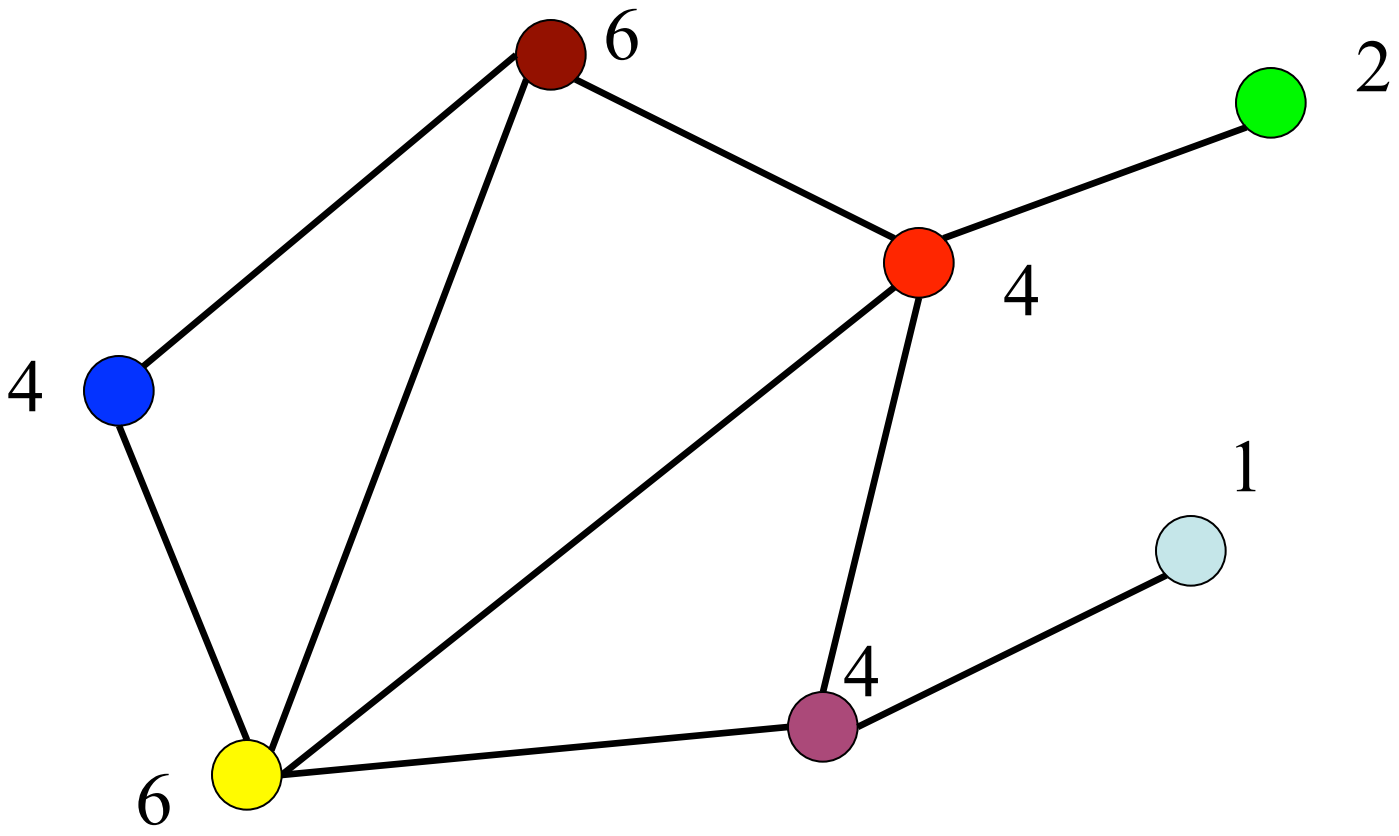
$V_{i+1} \leftarrow V_i - W_i$

$i \leftarrow i+1$

$V_i \leftarrow V_i - D_i$ ($D_i = \{v \in G_i \mid \deg(v)=0\}$)

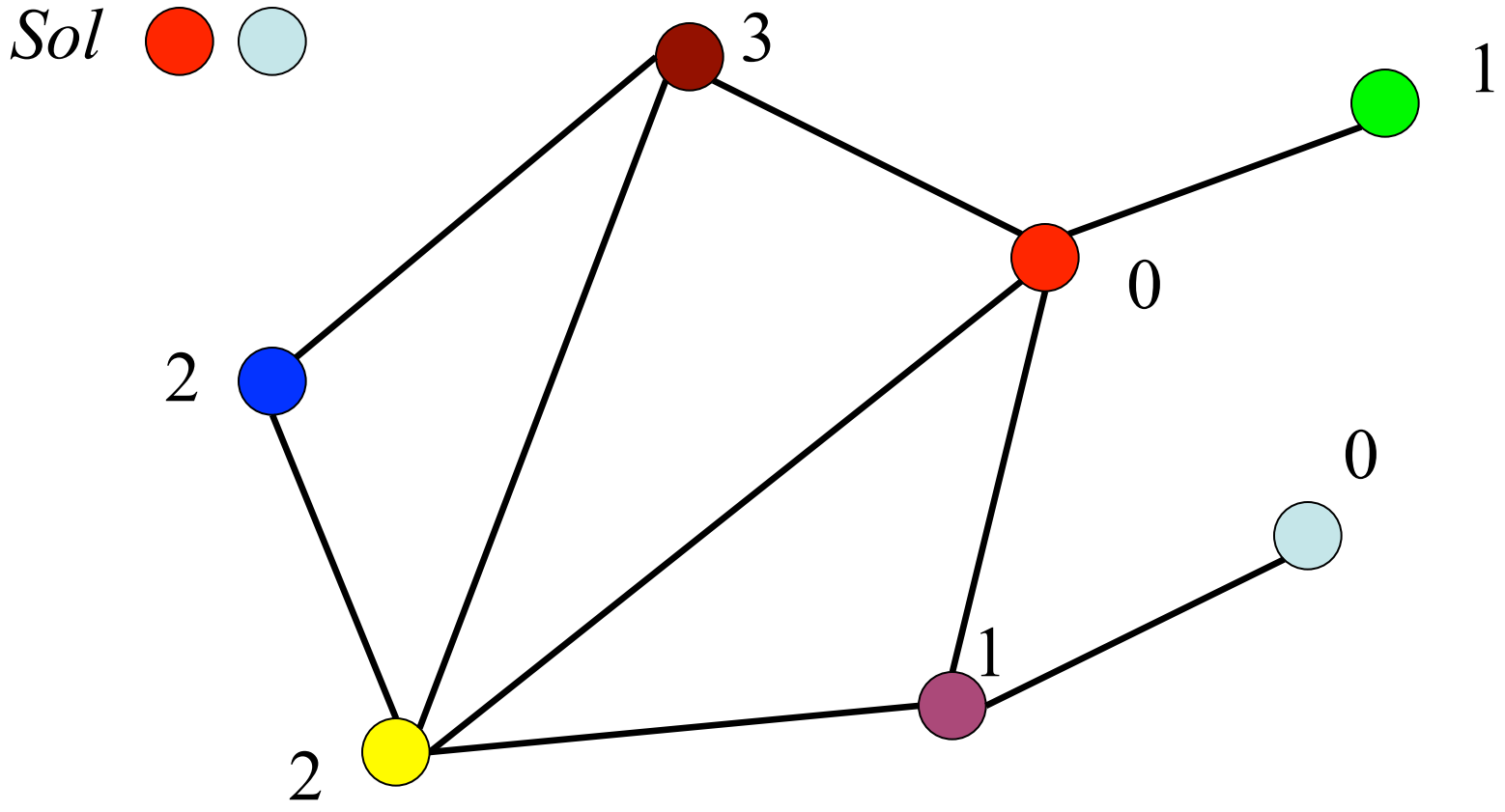
3) **Output** (Sol)

Пример

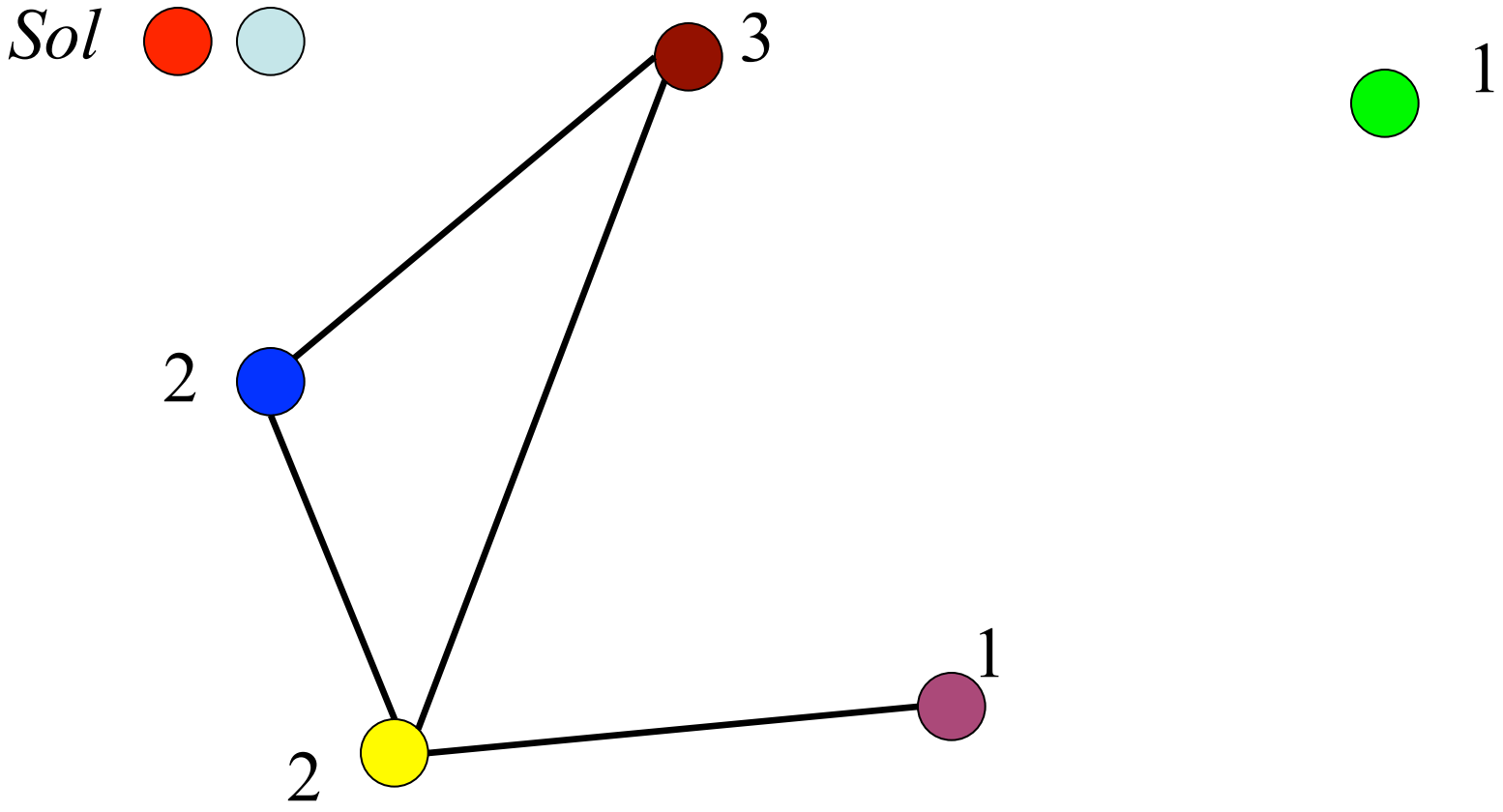


$$t(v) = 1 \cdot \deg(v)$$

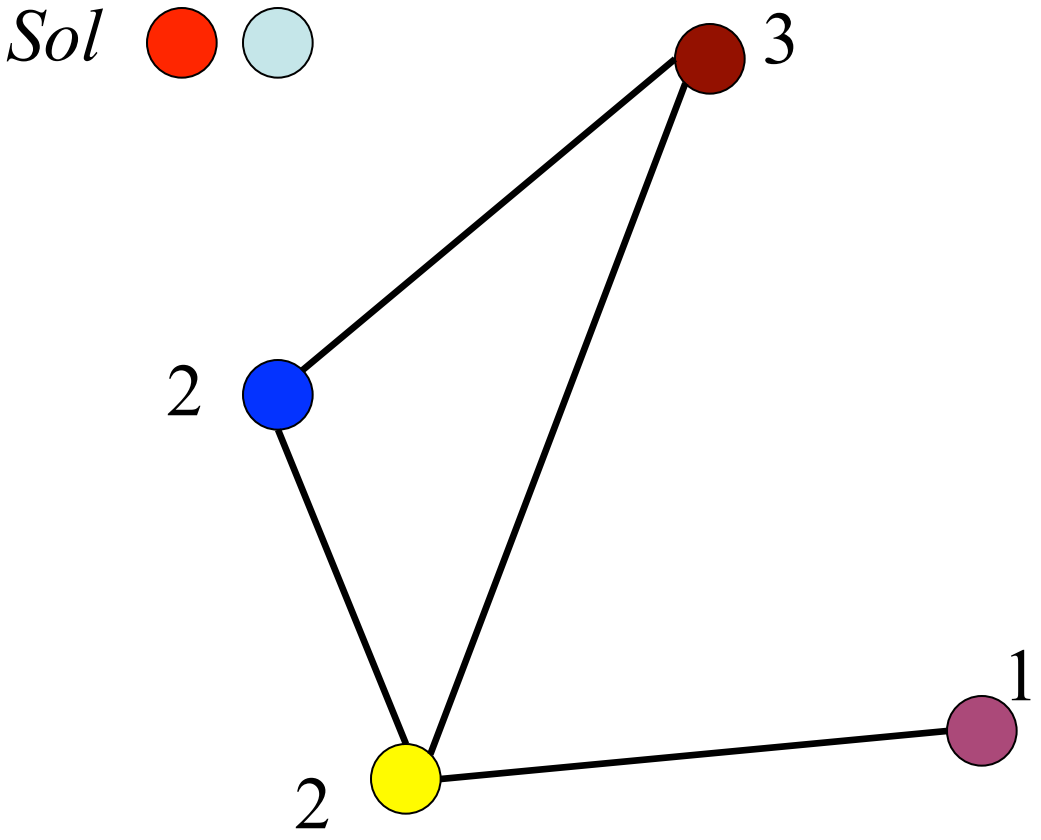
Пример



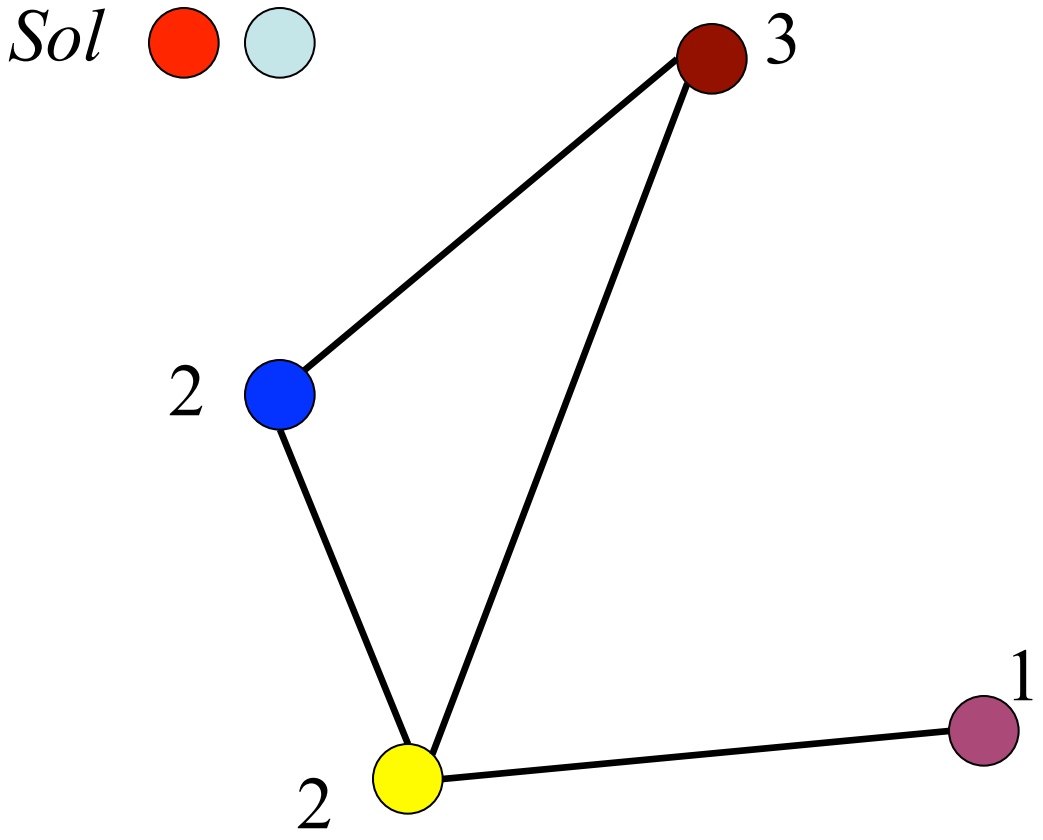
Пример



Пример

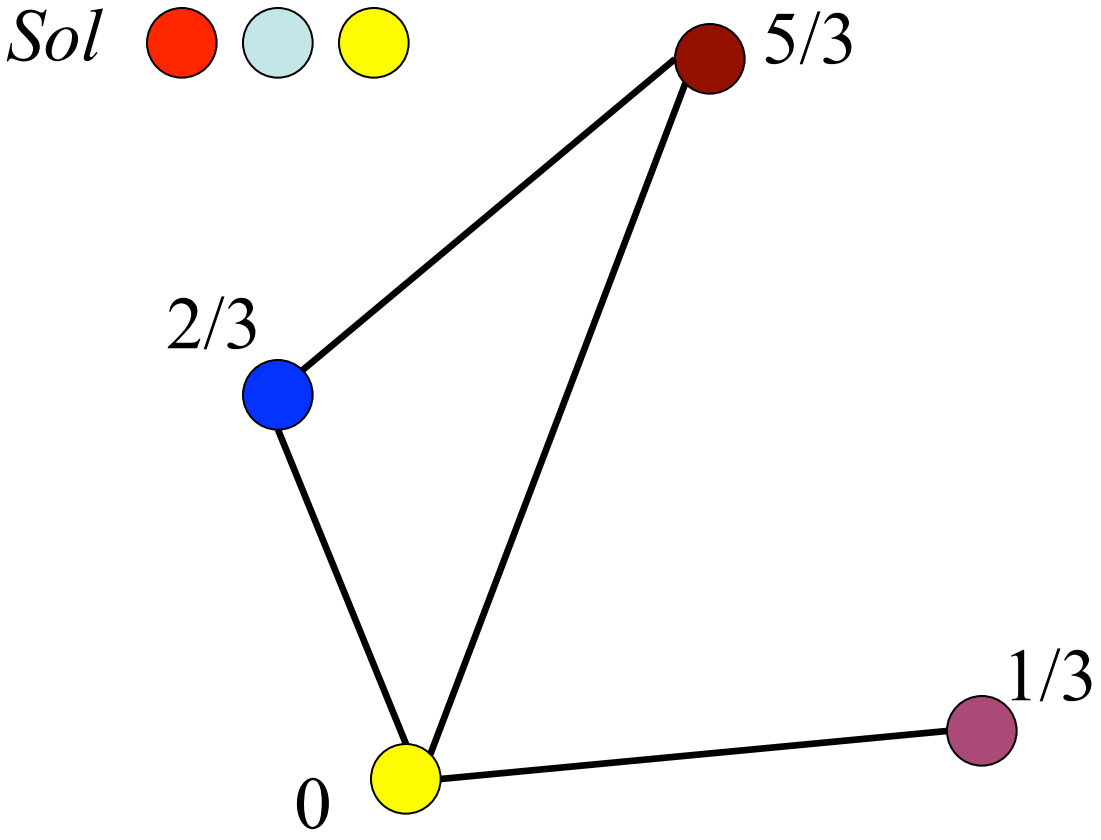


Пример

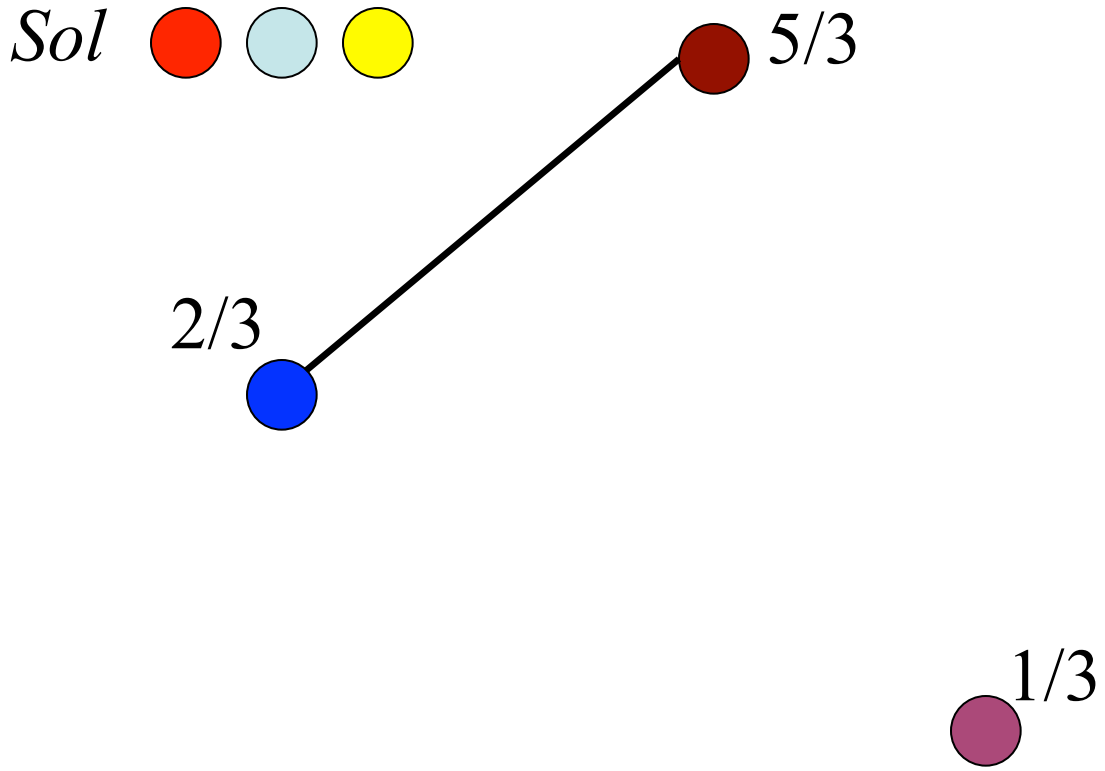


$$t(v) = (2/3) \cdot \deg(v)$$

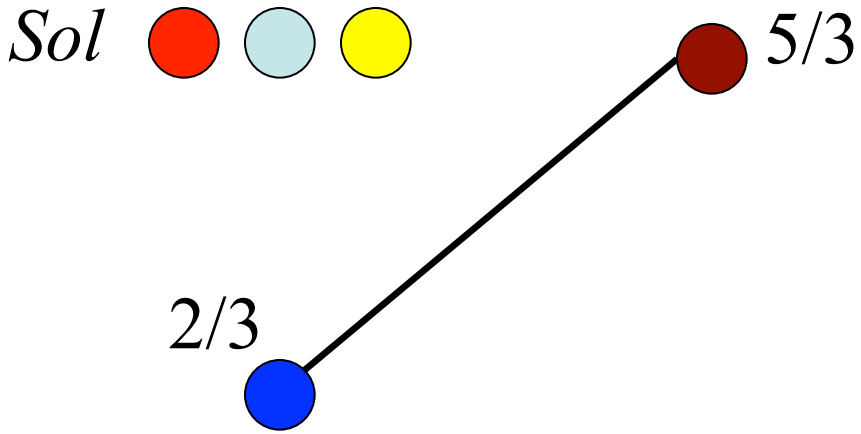
Пример



Пример

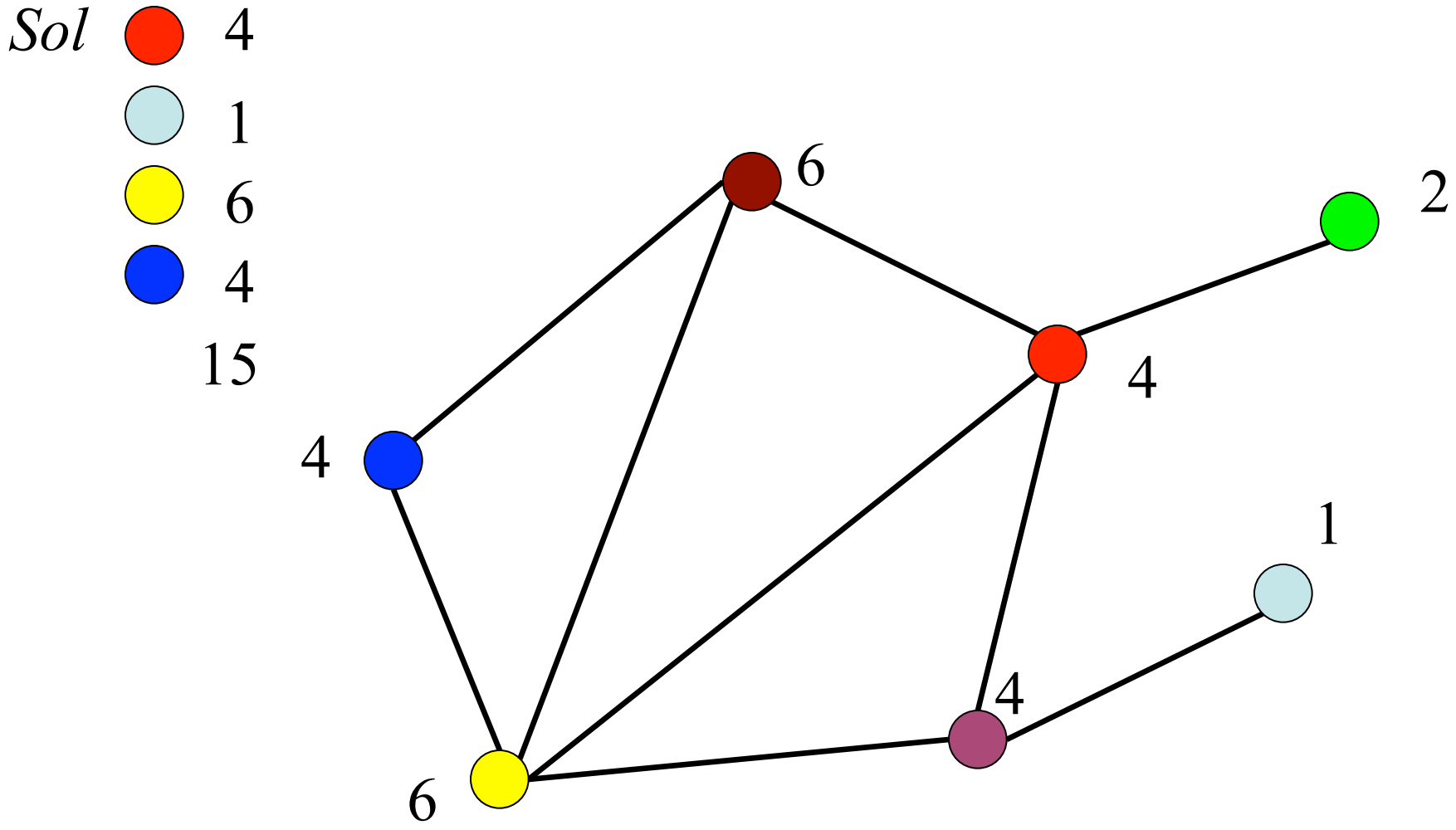


Пример



$$t(v) = (2/3) \cdot \deg(v)$$

Пример



Оценка качества алгоритма «Уровневый»

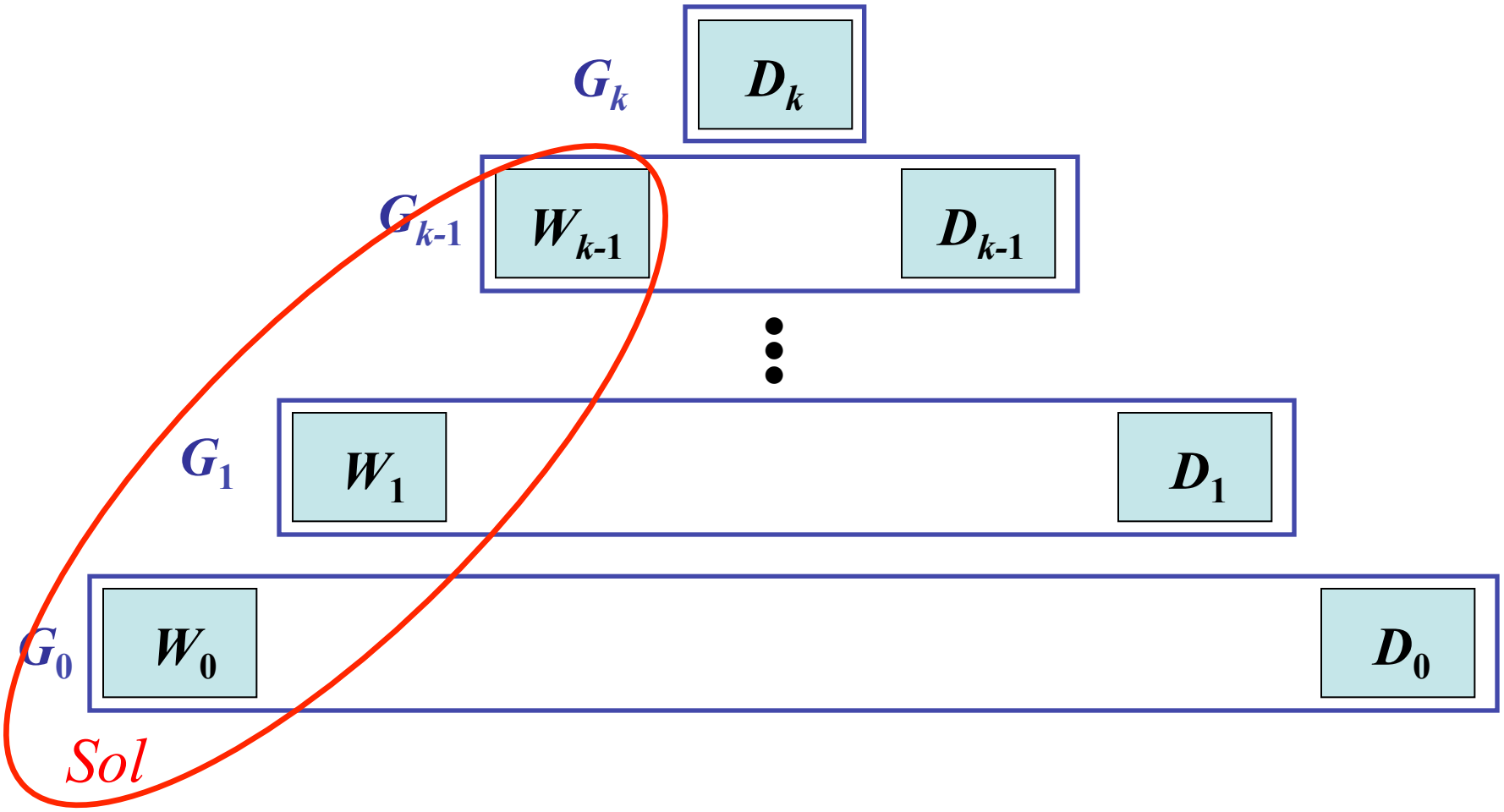
Теорема 2.4

Алгоритм «Уровневый» является 2-приближенным алгоритмом для задачи о вершинном покрытии.

Доказательство

- Пусть $t_0,\ldots,t_{k-1}$– пропорционально-степенные функции на графах $G_0,\ldots,G_{k-1}$.
- Тогда $Sol = W_0 \cup \ldots \cup W_{k-1}$ и $V \setminus Sol = D_0 \cup \ldots \cup D_k$.

Схема работы алгоритма



Доказательство корректности

$Sol = W_0 \cup \ldots \cup W_{k-1}$ $\qquad$ $V \setminus Sol = D_0 \cup \ldots \cup D_k$.

Покажем, что множество *Sol* является вершинным покрытием для графа *G*.

- Если это не так, то существует ребро (u,v) такое, что $u \in D_i$ и $v \in D_j$.
- Пусть $i \le j$.
- Тогда (u,v) должно лежать в G_i.
- Это противоречит предположению, что степень u равна нулю.

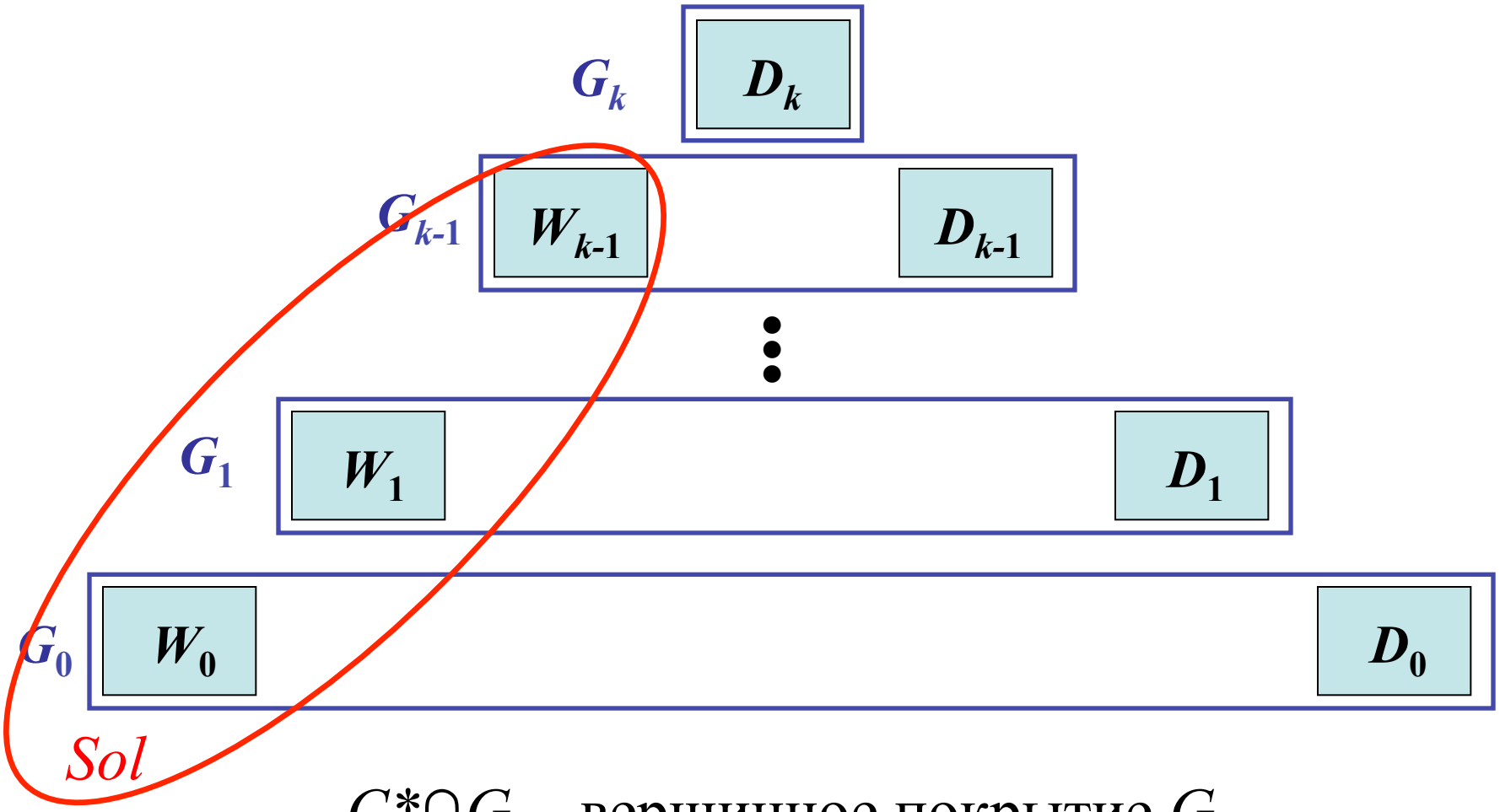
Доказательство оценки

- Пусть C^* – оптимальное вершинное покрытие.

$$v \in W_j : \quad w = \sum_{i \le j} t_i(v)$$

$$v \in D_j : \quad w \ge \sum_{i < j} t_i(v)$$

Схема работы алгоритма



$C^* \cap G_i$ – вершинное покрытие G_i

Доказательство оценки

- Пусть C^* – оптимальное вершинное покрытие.

$$v \in W_j : \quad w = \sum_{i \le j} t_i(v)$$

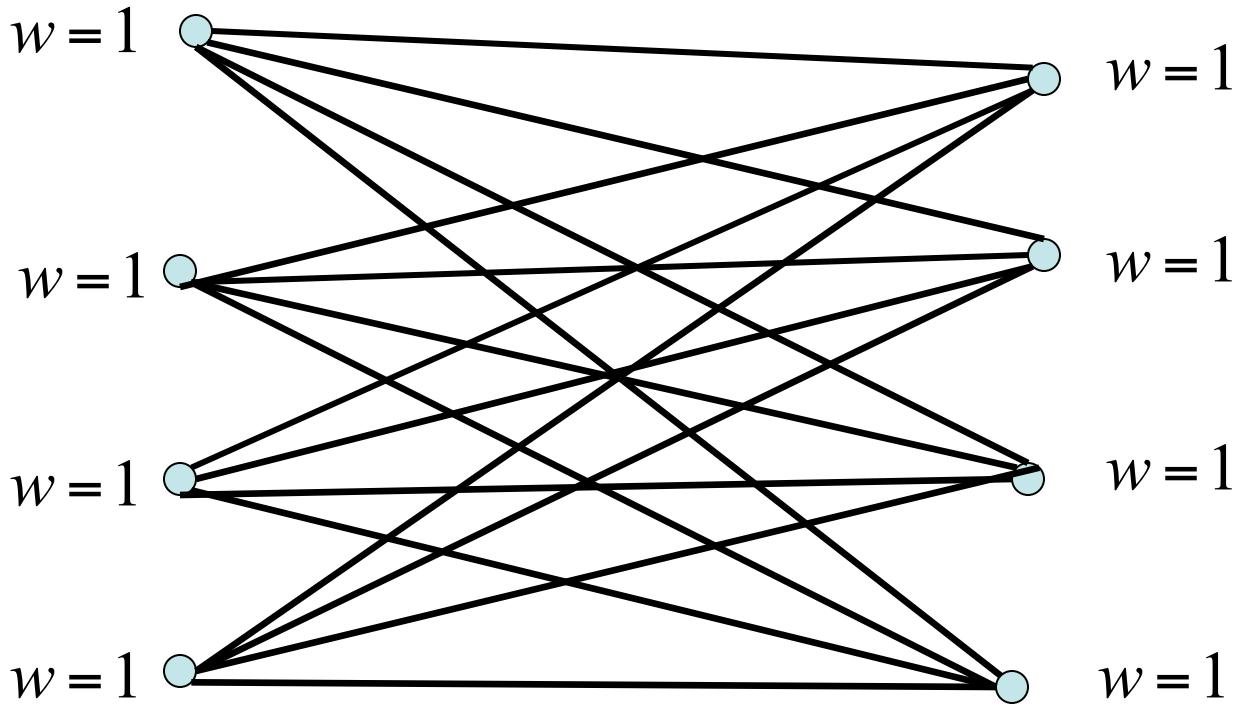
$$v \in D_j : \quad w \ge \sum_{i < j} t_i(v)$$

$C^* \cap G_i$ – вершинное покрытие G_i

$$t_i(Sol \cap G_i) \le t_i(G_i) \le 2 t_i(C^* \cap G_i)$$

$$w(Sol) = \sum_{t=0}^{k-1} t_i(Sol \cap G_i) \le 2 \sum_{t=0}^{k-1} t_i(C^* \cap G_i) \le 2 w(C^*)$$

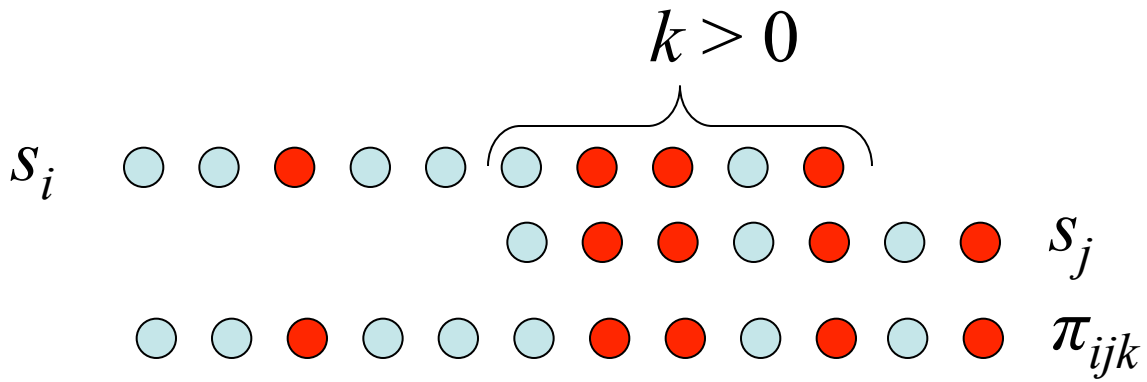
Точность оценки



Задача о кратчайшей суперстроке

- *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.

Задача о кратчайшей суперстроке как задача о покрытии множествами



$M = \{\pi_{ijk} \mid \pi_{ijk} - \text{допустимая}\}$

$\pi \in M: \quad \text{set}(\pi) = \{s \in S \mid s - \text{подстрока } \pi\}$

$U_{\text{cover}} \equiv S_{\text{string}}$

$S_{\text{cover}} \equiv \{\text{set}(\pi_{ijk}) \mid \pi_{ijk} - \text{допустимая}\}$

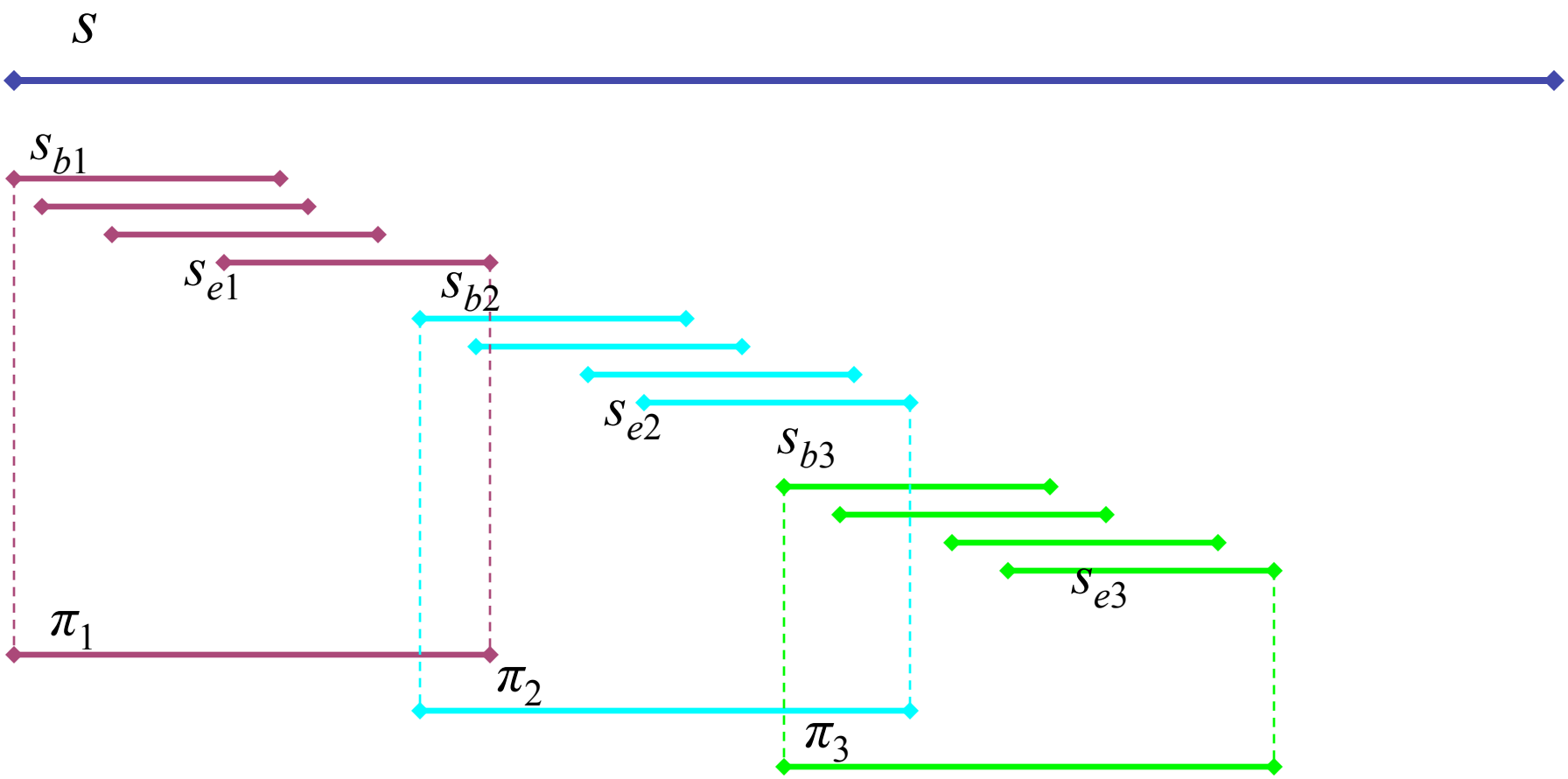
$c(\text{set}(\pi)) = |\pi|$

Нижняя оценка

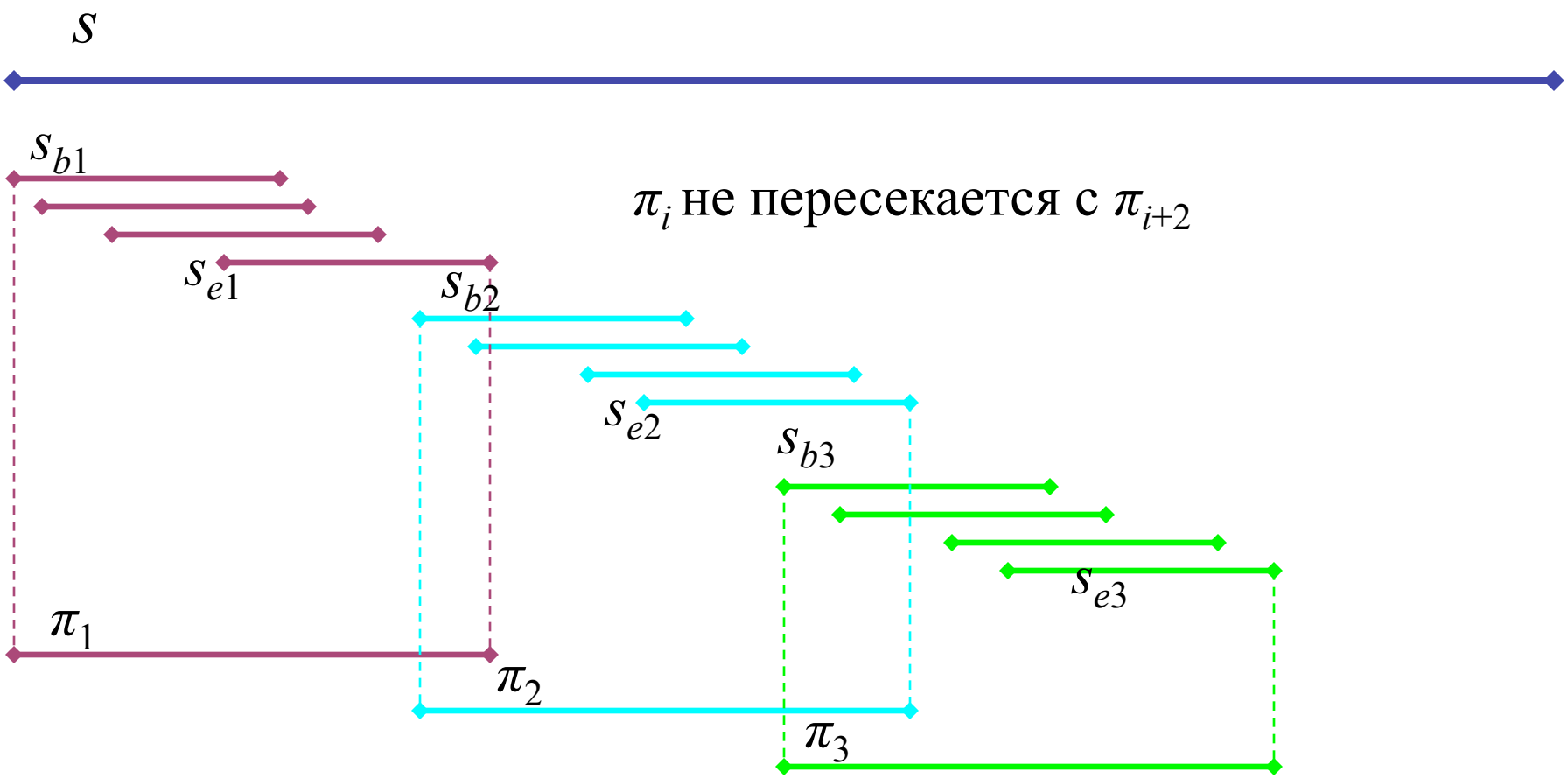
- **Лемма 2.5**

$$OPT_{string} \leq OPT_{cover} \leq 2\, OPT_{string}$$

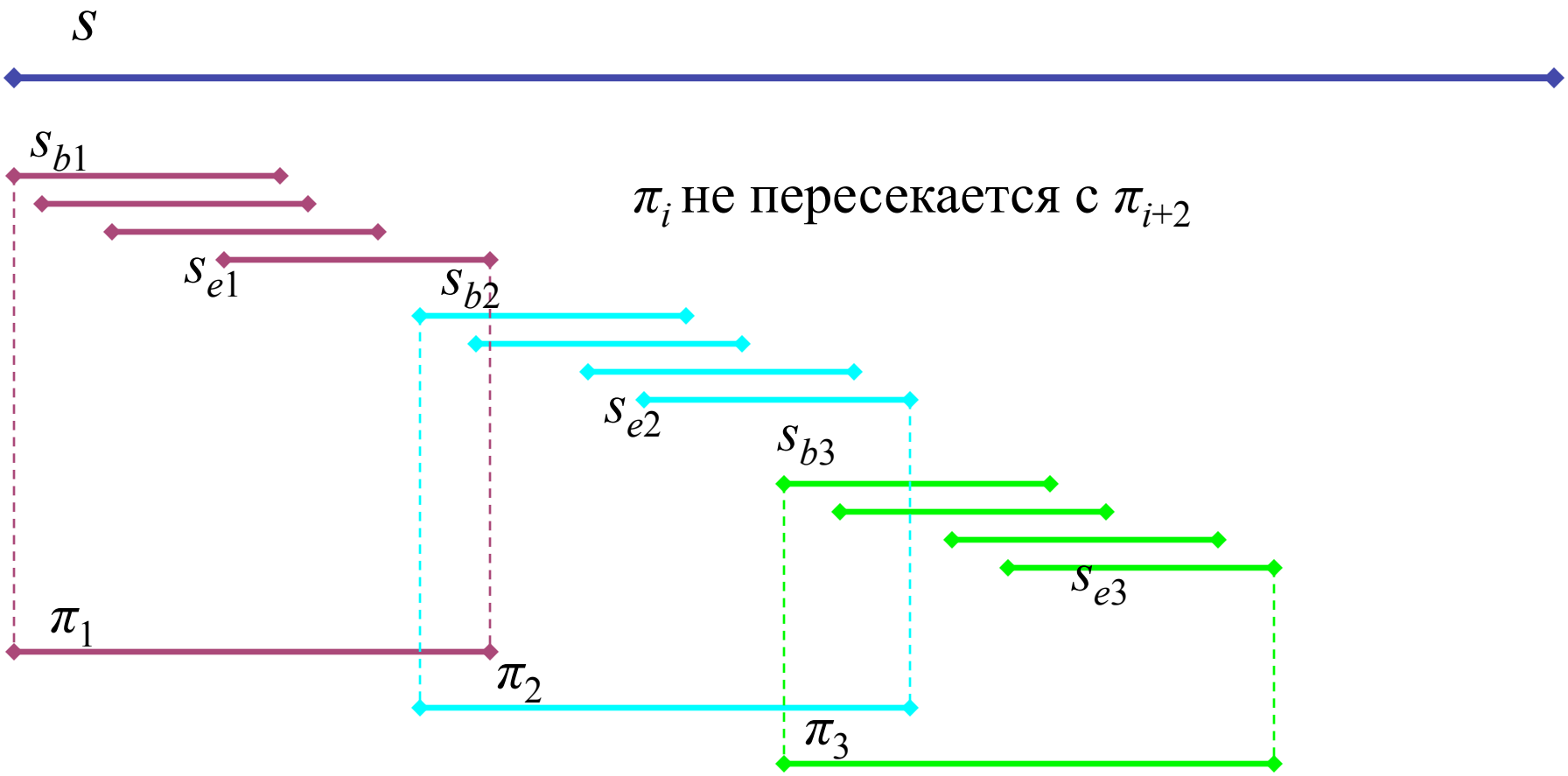
$OPT_{cover} \leq 2\ OPT_{string}$



$OPT_{cover} \leq 2\ OPT_{string}$



$OPT_{cover} \le 2\, OPT_{string}$



π_i не пересекается с π_{i+2}

$\{\text{set}(\pi_i) | i=1,\cdots,t\}$ допустимое решение со стоимостью $\sum_i \pi_i$.

Алгоритм Ли

1) По индивидуальной задаче о кратчайшей суперстроке строим индивидуальную задачу о покрытии множествами.
2) Алгоритмом Хватала находим покрытие.
   Пусть оно состоит из множеств set(π_1),…, set(π_k).
3) Соединяем строки π_1,…,π_k в любом порядке. Назовем полученную строку s.
4) **Output** (s)

Оценка качества алгоритма Ли

Теорема 2.6

Алгоритм Ли является $2H_n$ -приближенным алгоритмом для задачи о кратчайшей суперстроке, где n – число строк в исходном примере.

Практика

Задача об упаковке в контейнеры с ограничением на число предметов в одном контейнере.

- *Дано*: n предметов и их размеры $a_1,\ldots,a_n \in (0,1]$.
- *Найти* упаковку всех предметов в единичные ящики, так чтобы минимизировать число использованных ящиков, при условии что каждый ящик содержит не более пяти предметов.

1. Сформулировать задачу об упаковке как задачу о покрытии.
2. Является ли ваше сведение полиномиальным.

Практика

Рассмотрим следующую жадную стратегию для задачи о вершинном покрытии наименьшей мощности. Возьмем в решение произвольную вершину максимальной степени и удалим ее вместе с инцидентными ей ребрами. Продолжим эту процедуру до тех пор, пока в графе не останется ребер. Докажите, что полученное решение не превосходит $H_n \cdot \mathrm{OPT}$.