

NP-трудные задачи

Задача поиска

Пространство поиска $\sim 2^n$

≡ Задача поиска (search problem)

$C(I, S)$ - предикат от двух параметров

I - (instance) условие

S - (solution) решение

Вход: I $C(I, S) = \text{true} \Leftrightarrow$

Выход: S S - решение для I

Мы будем рассматривать задачу поиска,

где C - вычисляется за $O(\text{poly}(|I| + |S|))$

≡ Задача разрешения (decision problem)

$C(I, S)$ - " -

Вход: I

Выход: $\exists? S : C(I, S) = \text{true}$

≡ Задача выполнимости (Satisfiability)

(SAT)

Вход: формула в КНФ

$$\varphi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_4 \vee x_5 \vee x_6 \vee \overline{x_1}) \wedge \dots \wedge (x_2 \vee \overline{x_4})$$

переменная

литерал

гизьонка / клоз
(clause)

(Search): найти значения x_1, \dots, x_n

$$\varphi(x_1, \dots, x_n) = \text{true}$$

выполнимый набор

(Decision): $\exists? x_1, \dots, x_n : \varphi(x_1, \dots, x_n) = \text{true}$

Лемма: Можно решить decision problem где SAT за полиномиальное время \Leftrightarrow можно решить и search problem за полиномиальное время.

\Leftarrow очевидно

$\Rightarrow \exists A: A(\varphi) = 1 \Leftrightarrow \exists x_1 \dots x_n: \varphi(x_1 \dots x_n) = \text{true}$
(A - полиномиальный алгоритм)

B: $A(\varphi) \rightarrow 1$

$\searrow 0 \Rightarrow B(\varphi) = \text{нет решения}$

$A(\varphi [x_1 = \text{true}]) = 1 \rightarrow A(\varphi [x_1 = \text{true}, x_2 = \text{false}])$
 $A(\varphi [x_1 = \text{false}])$ $A(\varphi [x_1 = \text{true}, x_2 = \text{true}])$

\downarrow

Время работы B:

$$\text{time}(B(\varphi)) = \text{time}(A(\varphi)) \cdot n$$

NB: алгоритм - эффективный \Leftrightarrow работает $\text{for } O(\text{poly}(n)) = n^{O(1)}$.

Что известно про SAT?

- Не удается решить быстрее $O(2^n \text{poly}(n))$
- Удается решить k-SAT быстрее: $O(2^{\epsilon n})$ $\epsilon < 1$
и при $k \geq 3$
- При $k=2$ - решается $\text{for } O(|\varphi|)$
- Хорошие формулы:

в # гурьянские ≤ 1 черемной
дег отрицание
да $O(|V|)$

TSP (travelling salesman problem,
коммивояжера)

Вход: полный взвешенный граф

Выход: Обход всех всех вершин графа
минимальной стоимости

Аналог: MST

Hamiltonian Cycle (Хам, Рудрата cycle)

Вход: граф

Выход: циклический обход всех вершин
графа (# вершин только ≥ 1 раз)

\sim Hamiltonian Path

// Euler cycle

Balanced Cut

// Min Cut

Вход: взвешенный граф

Выход: Минимальный сбалансированный разрез

$$V - S \subseteq \bar{S}$$

$$\frac{1}{3}|V| \leq |E| \leq \frac{2}{3}|V|$$

Integer Linear Programming // LP

Вход: A, b, c

Выход: $x: Ax \leq b, \quad c^T x \rightarrow \max$

$$x_i \in \mathbb{Z}$$

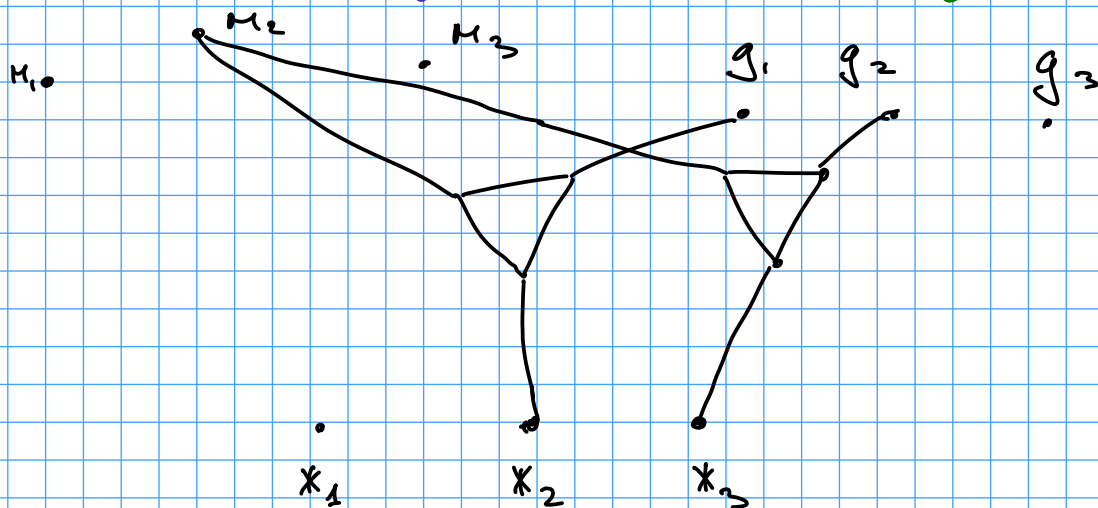
Zero-one equation (ZOE)

Вход: $A \in \{0, 1\}^{n \times n}$

Выход: $x: Ax = \bar{1}, \quad x \in \{0, 1\}^n$

3D-matching

// Matching



Вход: мультиграф с 3-рёбрами

Выход: всегда и непересекающиеся рёбра.

Independent Set // IS on trees

Вход: граф, число k

Выход: независимое множество размера k
(нет рёбер между вершинами этого множества)

Vertex Cover // VC on trees

Вход: граф, число k

Выход: вершинное покрытие размера k
(рёбро имеет один из концов в этом множестве)

Set Cover

Вход: $\{S_i\}$: $\bigcup S_i = U$, k

Выход: k S_i : $\bigcup_{j=1}^k S_{i_j} = U$

Clique

Вход: граф, число k

Выход: клика размера k

Longest Path // Shortest Path

Вход: граф, вершины s, t , число k

Выход: путь $s \rightarrow t$ длины k

Задача о рюкзаке // Уменьший рюкзак

SubSet Sum

Вход: $U \subseteq \mathbb{Z}$, k

Выход: $M \subseteq U : \sum_{a \in M} a = k$

Graph Coloring

Вход: граф, число k

Выход: раскраска графа

P vs NP

\equiv P - класс задач поиска, решаемых за $n^{O(1)}$ (\exists эффективный алгоритм)

\equiv NP - класс задач поиска, у которых предикат вычисляется за $n^{O(1)}$

Основной вопрос: $P \stackrel{?}{=} NP$

Известно: $P \subset NP$

Сложный вопрос: $coNP \stackrel{?}{=} NP$

$NP \neq coNP \Rightarrow P \neq NP$

Утв. Все уменьшит. задачи из класса NP

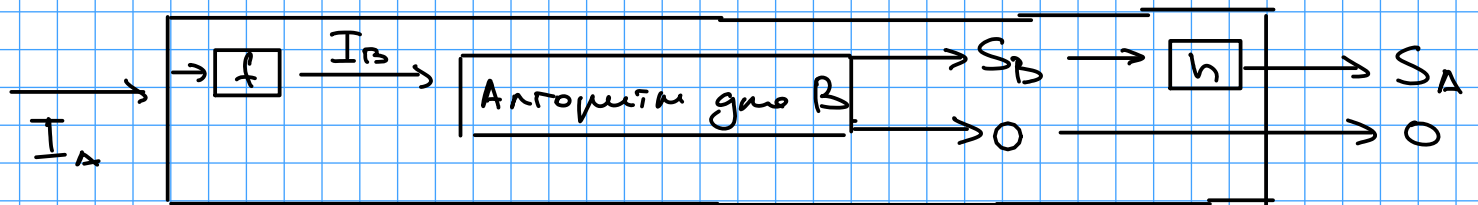
Сведение

$A \rightarrow B$ // сведение по Копру,
мног.-оное сведение

\exists поли. введ. функции:

$$f: \{I_A\} \rightarrow \{I_B\}$$

$$h: \{S_B\} \rightarrow \{S_A\}$$



- $\forall I_A \exists S_A: C_A(I_A, S_A) \Leftrightarrow \exists S_B: C_B(f(I_A), S_B)$
- $\forall I_A \forall S_B: C_B(f(I_A), S_B) \Rightarrow C_A(I_A, h(S_B))$

Сб-Св:

- B решает $f \circ h^{-1}(A) \Rightarrow A$ решается $f \circ h^{-1}(A)$
- $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$

\equiv NP - трудная задача

Задача к которой сводится \forall задача из класса NP

\equiv NP - полная задача

NP - трудная задача из класса NP (класс NPC)

Будут сложные задачи в NP, про которые не известно, что они NPC

- Factoring

Вход: n

Выход: $p, q: p \cdot q = n, p, q \geq 1$

- Graph Isomorphism

Вход: G_1, G_2

Выход: π - перест. вершин: $\pi(G_1) = G_2$

Сведение:

SAT \rightarrow 3SAT

SAT: $(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \dots \vee x_k) \wedge \dots$

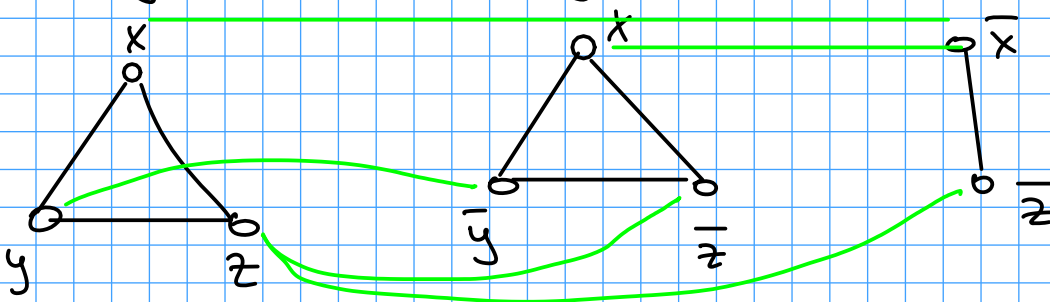
3SAT: $(x_1 \vee x_2 \vee y_1) \wedge (\bar{y}_1 \vee x_3 \vee y_2) \wedge (\bar{y}_2 \vee x_4 \vee y_3) \wedge \dots$
 $\wedge \dots (\bar{y}_{k-4} \vee x_{k-2} \vee y_{k-3}) \wedge (\bar{y}_{k-3} \vee x_{k-1} \vee x_k)$

$\sim k$ кругов

↑
описание f

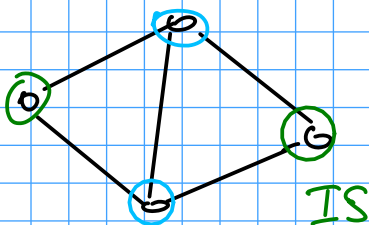
3SAT \rightarrow IS

$(x \vee y \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{z})$



Ищем IS размера # кругов

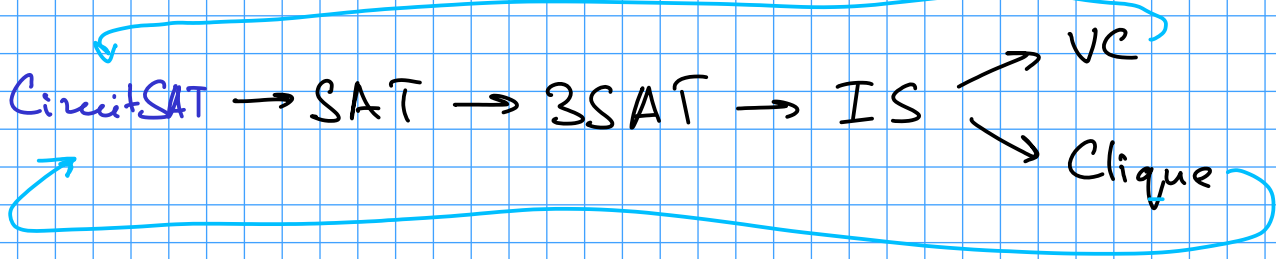
IS \rightarrow VC



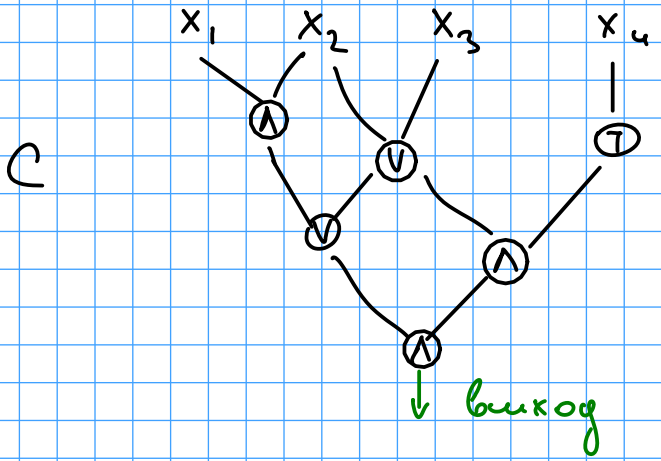
$IS((V, E), k) \rightarrow$
 $VC((V, E), |V| - k)$

IS \rightarrow Clique

$IS((V, E), k) \rightarrow \text{Clique}((V, \bar{E}), k)$

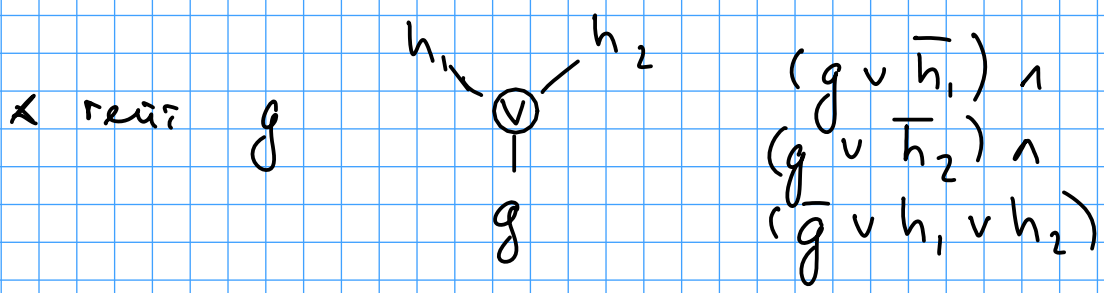


Circuit SAT



Выход: схема C
 Выход: такие x_i !
 $C(x_1, \dots, x_n) = 1$

Circuit SAT \rightarrow SAT



Теорема (Куна - Левина)

\forall NP задача \rightarrow Circuit SAT

$\exists X \in NP$

$\exists C_x$ - предикат: $C_x(I, S) = 1$ решение I
 C_x вычисляется за полиномиальное время на "компьютере"

Δ программу дать C_x как схему
 Построим схему для C_x и искомого условия $I \in X$

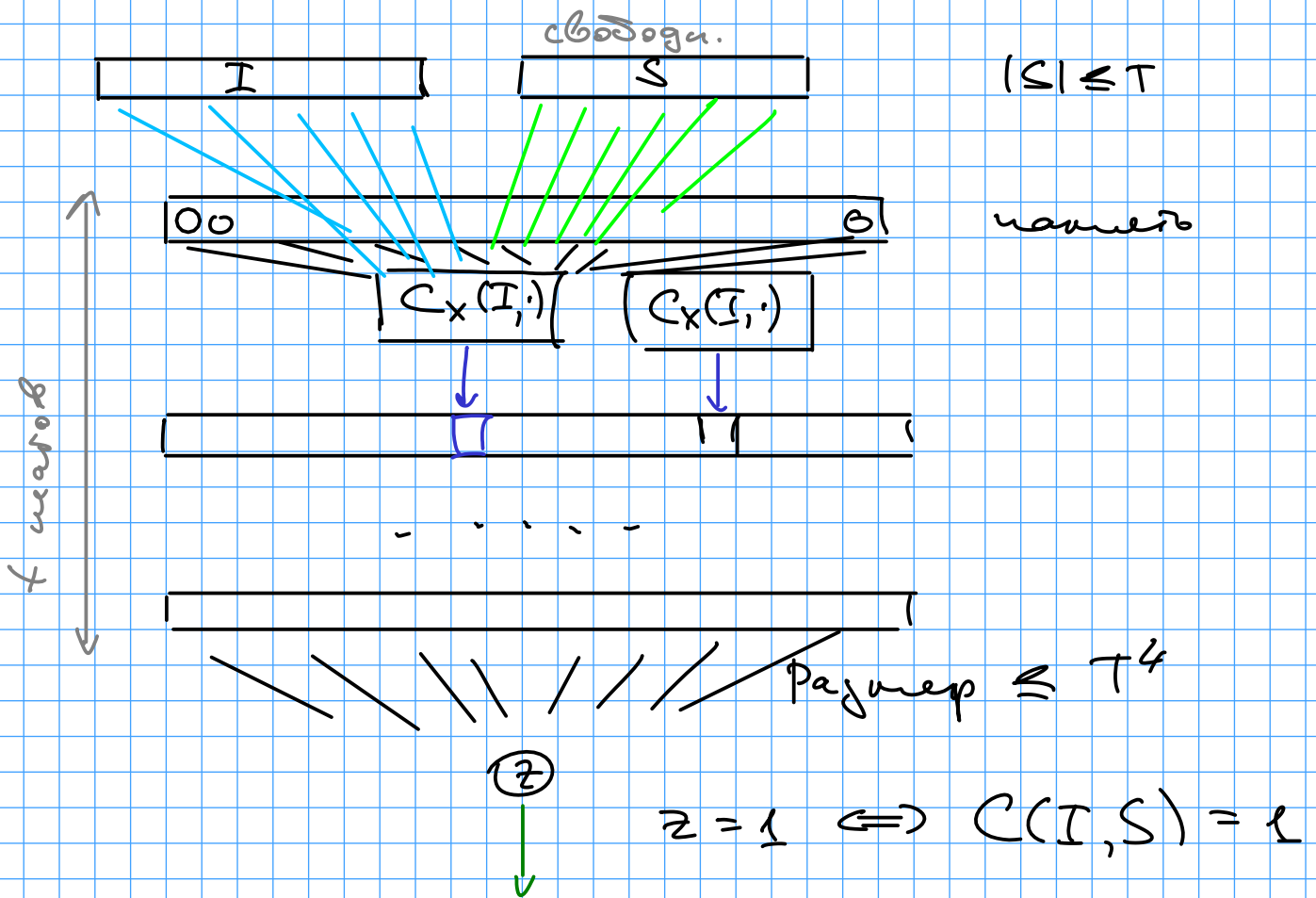
t.e. схема gate $C_x(I, \cdot)$

↑
свободные
переменные

C_x лемма о том же $t(n) \in poly(n)$

$C_x(I, \cdot)$ будет работать же $t(T) = T$
времем

Как будет устроена схема



\forall NP задача \rightarrow схема $C(I, \cdot)$

Circuit SAT

Следствие 1: Circuit SAT - NP полная задача

Следствие 2: SAT, 3SAT, IS, VC, Clique ...

Тогда NP - полная (NP complete)

Общий вопрос

Стоит ли верить, что $P \neq NP$?

Почему NP?

P = polynomial time

NP = non-deterministic poly. time

$$\exists y M(x, y) = 1$$

Как решать NP-трудные задачи?

1. Brute force
2. Эвристический
 - перебор с возвратом
 - метод "ветвей и границ"
 - локальный поиск

SAT Solvers - очень хорошо развитая область

3. Приближенные алгоритмы

Set Cover

"Жадная" стратегия даёт $\log(N)$ приближ.

Vertex Cover

$\log(N)$ приближение жадной стратегией

2 приближение:

1. Найдём макс. по величине независимое множество (жадным образом)

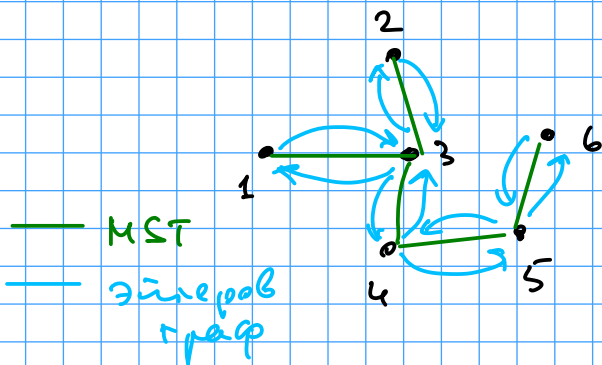
2. Возьмем $S = \{ \text{все вершины из графа} \}$

УТВ: $S = VC$

УТВ: $|S| \leq 2 |MST|$

Метрический Коммивояжер (metric TSP)

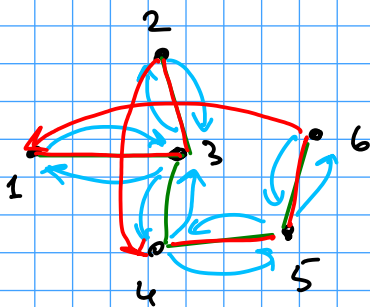
Выполняется неравенство Δ .



$\exists S$ - решение задачи TSP

$|S| \geq MST$

Пусть это Эйлеров обход длины $2MST$



Давайте упростим обход, чтобы он заходил в \forall вершину = 1 раз

1 3 2 4 5 6 1

Длина полученного обхода $\leq 2MST \leq 2|S|$

NB: для обычного TSP оптимального приближения не \exists (если $P \neq NP$)
доказ-во: сведите Гамильтонов цикл.