8 сентября 2021 г. 10:32

Exact Exponential Algorithms

NP

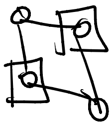
• точные алгоритмы

YES/NO

↑ Decision

Vertex Cover

$\leq k$



SETH

$(2-\varepsilon)^n \cdot poly(n)$???

число переменных

• Все алг-мы работают $\Omega(c^n)$

$c > 1$

NP-трудными (NP-полные)

$poly(n) \cdot 2^n$ ← Satisfiability CNF $(\vee\vee\vee)\wedge(\vee\vee\vee)\wedge\ldots\wedge(\vee\vee\vee)$

Vertex Cover $\to 2^n$

$\to 1.1996^n$

Travelling Salesperson Problem

$n! \cdot poly(n) \sim 2^{n \log n}$ $\quad c^n$

$O(2^n \cdot n^{200}) \to O^*(2^n)$

$n^{1.5}$

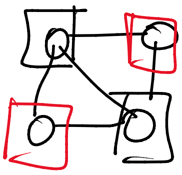
Опр $f(n) = O^*(g(n)) \Leftrightarrow \exists c: f(n) = O(g(n) \cdot n^c)$

$c^n \cdot n^k = O((c+\varepsilon)^n)$ $\quad \frac{(c+\varepsilon)^n}{c^n}$

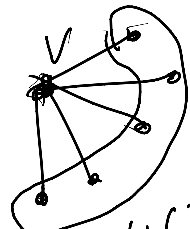
$\boxed{\text{Vertex Cover}}$

Ind Set

В G нез. мн-во р-ра $k \Leftrightarrow$ верш. покрытие р-ра $n-k$



2^n



mVC $\quad$ • $N(v)$

соседи v

v берем в верш. покрытие

Удаляем вершину v и
$mVC(G \setminus \{v\}) + 1$

v не берем в верш. покрытие

Берем всех ее соседей

mVC — • $N(v)$ — соседи v

• $\deg(v)$ — степень в-ны v

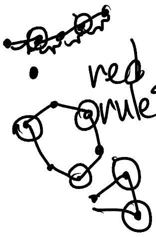
• $N[v]$ — соседи v + v

v не берем в верш. покрытие

[Удаляем верш. v и всех ее соседей + берем их в ответ

[$mvc(G \setminus N[v]) + \deg(v)$

$mvc(G)$:



• red. rule

[если в G $\Delta(G) \le 2$

вернуть $\sum \lceil E(C_i)/2 \rceil$ по к.с. графа G

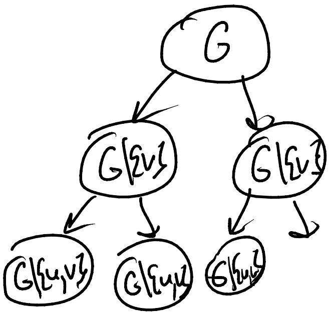
v ← вершина графа максимальной степени

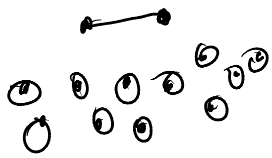
[вернуть $\min(mvc(G \setminus \{v\}) + 1,\ mvc(G \setminus N[v]) + \deg(v))$

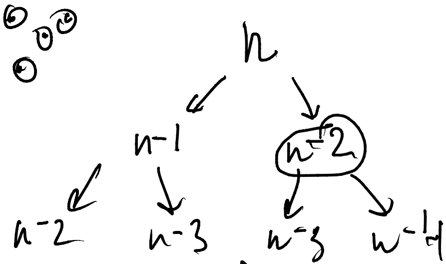
branch rule

Утв $mvc(G)$ возвр. размер min v.c. G.

$T(n) \le T(n-1) + T(n-3)$







$|N[v]| \ge 2$

$T(0) = 1$

$T(1) = 1$

$T(n) \le T(n-1) + T(n-2)$

1.618....

$T(n) = Fib_n \le 1.619^n$

<u>Branching</u> (расщепление) (ветвление)

$T(n) \ge T(n-1) + T(n-3)$ c^n

$$\begin{cases} T(0) = 1 \\ T(1) = 1 \\ T(2) = 1 \end{cases}$$

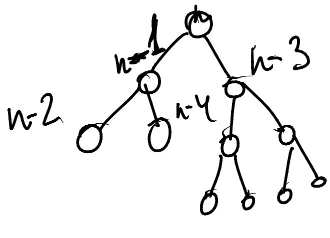
$T(n) = c^n$

Найти такое $c \ge 1$, что

$c^n \ge c^{n-1} + c^{n-3} \mid : c^{n-3}$

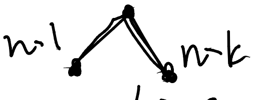
$c^3 \ge c^2 + 1$

$\underbrace{c^3 - c^2 - 1}\ \ge 0$



• T возрастающей

• T положительной



$k \ge 3$

$T(n) = T(n-1) + T(n-k) \le T(n-1) + T(n-3)$

$c \geq 1$

$c^3 - c^2 - 1 = 0$

$x \leq 1.466$

$T(n) \geq T(n-1) + T(n-3)$

1.466^n

$c^n \geq c^{n-1} + c^{n-2}$

$c^2 - c - 1 = 0$

Опр branching rule (правило ветвления)

poly время ← (Функция, которая из задачи р-ра n строит k задач р-ра $n-a_1, n-a_2, \ldots, n-a_k$ $\quad a_i \geq 1$ что ответ на задачу р-ра n находится из k ответов для подзадач

branching vector (вектор ветвления)

$(a_1, a_2, \ldots, a_k)$

branching constant (константа ветвл.)

$\tau(a_1, a_2, \ldots, a_k) =$ реш. уравнения $c^n - c^{n-a_1} - c^{n-a_2} - \ldots - c^{n-a_k} = 0$

$a_1 \leq a_2 \leq \ldots \leq a_k$ $\quad c^{a_k} - c^{a_k - a_1} - c^{a_k - a_2} - \ldots - 1 = 0$

Утв Если алг-м следует правилам ветвления из нек. набора B, то время работы алг-ма — $O^*(c^n)$, где c — наиб. константа ветвл. среди правил из B

$c^4 - c^3 - 1 = 0$ $\quad c \approx 1.3803$

$c^n - c^{n-1} - c^{n-4} = 0$ $\quad O^*(1.3804^n)$

1.38^n

$c^{(n)}$ $\quad c^{w_1 n_1 + w_2 n_2 + w_3 n_{\geq 3}}$

Measure & Conquer

Reduction rule

poly ← (Функция, которая преобразует задачу нек-м образом, что ответ на исх. задачу восст. из ответа на преобр. задачу poly ⟵ (функцию, ... что ответ на исх. задачу восст. из ответа на преобр. задачу

• Reduction rule должен упрощать (уменьшать нек. параметр)

Пример Удалить вершину ст. 0.

Утв Vertex Cover является NP-трудной на кубических графах (все ст. = 3)

danilka.pro@gmail.com

@dsagunov

$k=2$

k-SAT

3-SAT $(l_1 \vee l_2 \vee l_3) \wedge (l_4 \vee l_5 \vee l_6) \dots \wedge (l_8 \vee l_9)$ $2^{n^{0.9}}$

ETH нет $2^{\bar{o}(n)}$

Самый быстрый алгоритм для 3-SAT работает $\Omega(c^n)$ для нек. конст. c

$(l_1 \vee l_2 \vee l_3 \vee z) \wedge (\bar{z} \vee l_4 \vee \dots \vee l_k)$ $(x_1 \vee \bar{x}_1)$

k-SAT

$(l_1 \vee l_2 \vee l_3 \vee \dots \vee l_k)$ clause k-SAT(φ)

перебирать 1-й слева литерал, который выт. (?) клоз

0 0 1 $\le k$ переменных

перебираю 2^k означиваний этих переменных

RR1 Убрать все клозы, в которые входят два противоп. литерала

RR2 Убрать все переменные, кот. вх. только пол. или только отр.

RR3 Если возник пустой клоз, вернуть NO & все клозы, куда они входили

только $2^k - 1$ означивание возможно

$(\underbrace{k, k, k, \dots, k}_{2^k - 1})$ $\varphi|_{x_1=0\, x_2=1\, x_3=0}$

$2^k - 1$

k-SAT(φ): n – число перем.

RR1, RR2, RR3

берём клоз C, мн-во его переменных $x_1, x_2, \dots, x_k$

для каждого означивания $x_1, x_2, \dots, x_k$

если это озн. вып. клоз C

если k-SAT($\varphi|_{\text{подст. озн.}}$) вернула YES

вернуть YES

вернуть NO

$$c^n - c^{n-k} - c^{n-k} - \dots - c^{n-k} = 0$$

$$c^n - (2^k - 1)c^{n-k} = 0$$

$$c^k = 2^k - 1$$

$$c = \sqrt[k]{2^k - 1}$$

$\sqrt[3]{7} \leq 1.913$

$(\ell_1 \vee \ell_2 \vee \ell_3 \vee \dots \vee \ell_k)$

for $i = 1$ to k

$\ell_1, \ell_2, \dots, \ell_{i-1} \leftarrow 0$

$\ell_k \leftarrow 1$

k-SAT($\varphi|_{\dots}$)

$\dfrac{c^k - 1}{c - 1}$

$(1, 2, 3, 4, \dots, k)$

$(1, 2, 3, \dots, k-1)$

$$c^k - \underbrace{c^{k-1} - c^{k-2} - \dots - 1}_{} = 0$$

$$c^k - \frac{c^k - 1}{c - 1} = 0$$

$$\frac{c^{k+1} - c^k - (c^k - 1)}{c - 1} = 0$$

$$c^{k+1} - 2c^k + 1 = 0$$

c_k

1.6181^n

3-SAT → 1.8394^n

4-SAT → 1.9277^n

5-SAT

Утв k-SAT решается за $O^*(c_k^n)$

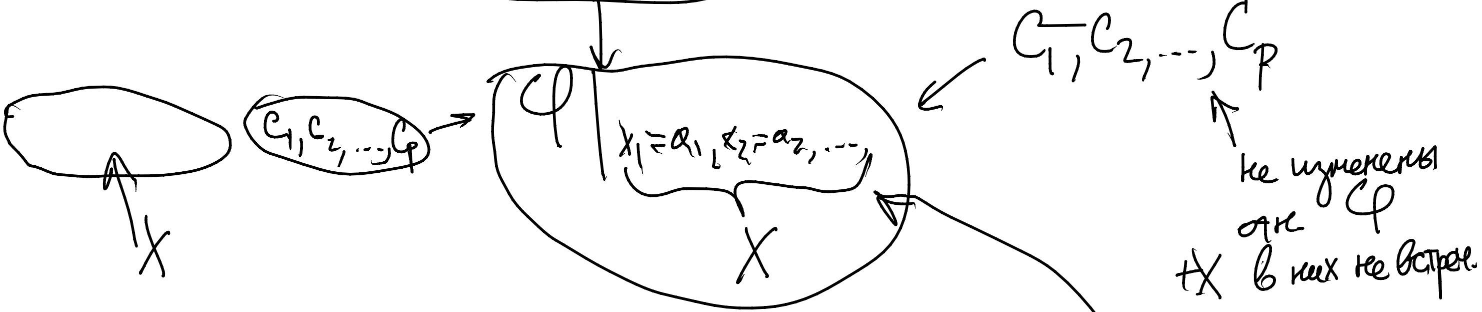
Утв k-SAT решается за $O^*(C_k^n)$

3-SAT

$O^*(C_{k-1}^n)$

$(1, 2, 3, \dots, k-1)$ ← ветвление происх. по клозу длины $k-1$

- Пусть в промежуточной формуле все клозы длины k



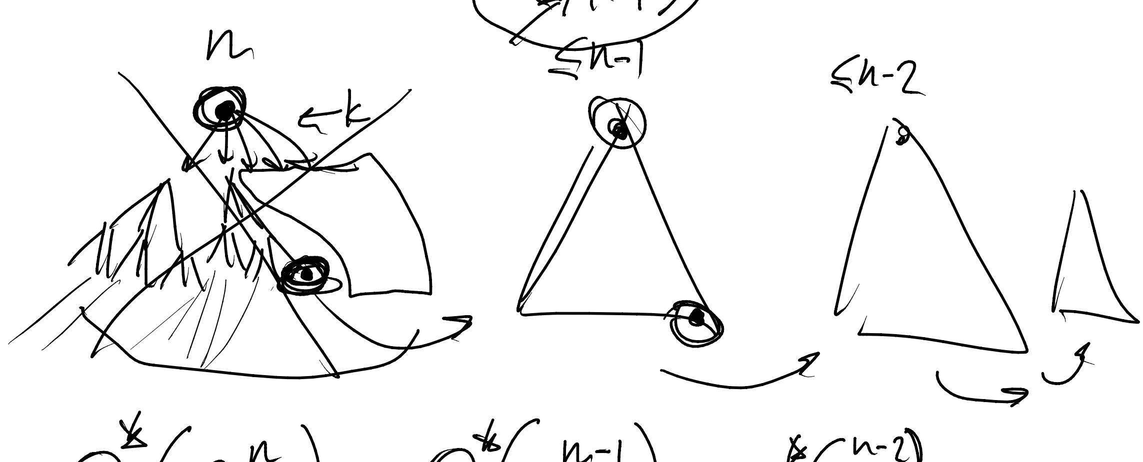
Утв Если φ выполнима, то $\exists$ такое означивание, которое вып. φ, при этом X в этом означивании

k-SAT$(\varphi|_{\dots})$

если в $\varphi|_{\dots}$ нет клозов длины $<k$, то решение всей исходной задачи k-SAT(φ) = k-SAT$(\varphi|_{\dots})$

φ

$\tau(1, 2, \dots, k-1)$



$O^*(C^n) \dots O^*(C^{n-1}) + O^*(C^{n-2})$

$$\underbrace{O^*(C_{k-1}^{n}) + O^*(C_{k-1}^{n-1}) + O^*(C_{k-1}^{n-2}) + \dots}_{O^*(C_{k-1}^{n})} \to$$

$\left(2-\frac{1}{k+1}\right)^n$ k-SAT

$\left(2-\frac{1}{4}\right)$ 1.75

$\left(2-\frac{1}{6}\right)$ $\frac{0.16666}{1.83333}$

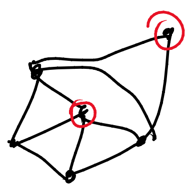
Vertex Cover ← задача

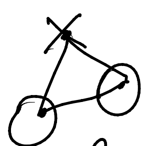
(G,k) ← экземпляр VC
instance

k-SAT ← задача

φ ← instance k-SAT

Пример Задача. дан граф G, нужно удалить из него $\leq k$ вершин, чтобы все треугольники исчезли



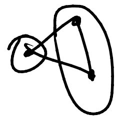
Возьмем любой треугольник 

8 вариантов $\sqrt[3]{7}$

triangle-free(G,k,P) ↑prohibited $\to (3,3,\dots,3)$ $(2,3,2)$

prohibited (4,4)

RRI Треугольник prohib. → вернуть NO



Задача реш. быстрее 2^n ⇒ SAT быстрее 2^n