LEC015 Max Coverage and Set Cover

VG441 SS2020

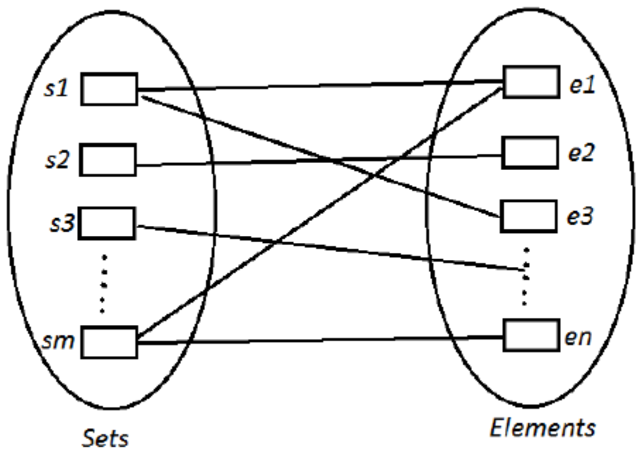
Cong Shi
Industrial & Operations Engineering
University of Michigan

Maximum Coverage

- A universe of elements $V = \{e_1, \ldots, e_n\}$
- A list of (possibly overlapping) sets $\{S_i \subseteq V\}_{i=1}^{m}$
- A bound K

Objective:

We wish to find K sets $S'_1, \ldots, S'_K$ such that $\left|\bigcup_{i=1}^{K} S'_i\right|$ is maximized



Greedy Algorithm

- The basic idea is to choose the set in each step which contains most of the uncovered elements

Input: V (set of all elements); $S_1, \ldots, S_n; K$

Output: Approximate solution $A_1, \cdots, A_K$ $U = V$

for $i = 1, \ldots, K$ do

  Let A_i be one of the sets $S_1, \ldots, S_n$ which maximizes $|A_i \cap U|$;

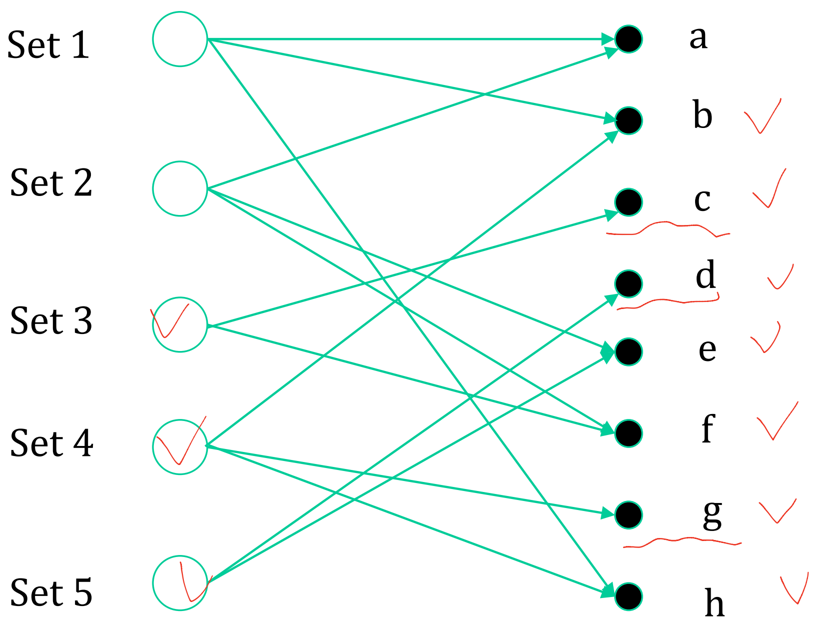
  $U = U \backslash A_i$;

end

An Example

- The basic idea is to choose the set in each step which contains most of the uncovered elements

Input: V (set of all elements); $S_1, \ldots, S_n; K$
Output: Approximate solution $A_1, \cdots, A_K$ $U = V$
for $i = 1, \ldots, K$ do
  Let A_i be one of the sets $S_1, \ldots, S_n$ which maximizes $|A_i \cap U|$;
  $U = U \backslash A_i$;
end

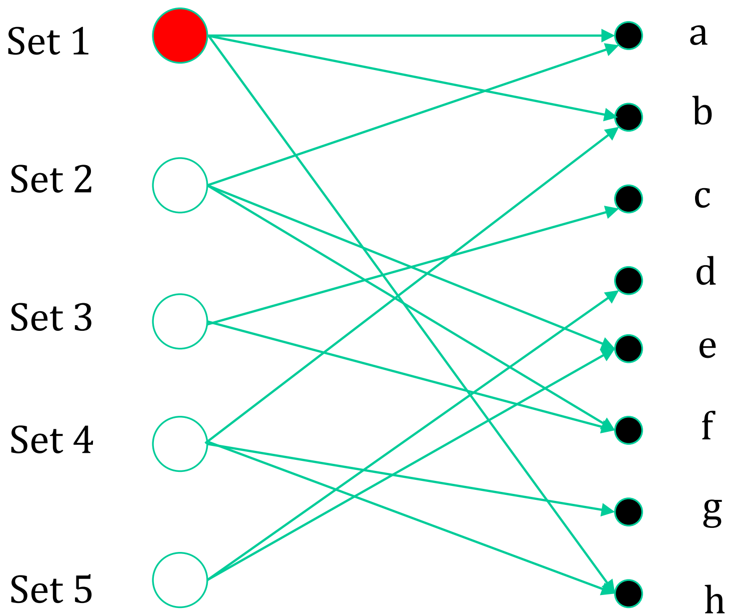


$K = 3$

An Example

- The basic idea is to choose the set in each step which contains most of the uncovered elements

Input: V (set of all elements); $S_1, \ldots, S_n; K$
Output: Approximate solution $A_1, \cdots, A_K$ $U = V$
for $i = 1, \ldots, K$ do
    Let A_i be one of the sets $S_1, \ldots, S_n$ which maximizes $|A_i \cap U|$;
    $U = U \backslash A_i$;
end

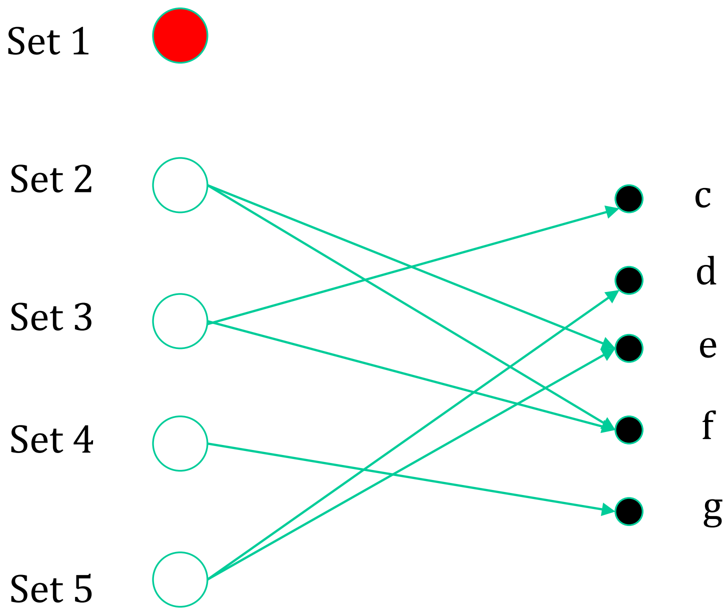


$K = 3$
Selected: Set 1, Covered Elements={a, b, h}

An Example

- The basic idea is to choose the set in each step which contains most of the uncovered elements

Input: V (set of all elements); $S_1, \ldots, S_n; K$
Output: Approximate solution $A_1, \cdots, A_K$ $U = V$
for $i = 1, \ldots, K$ do
  Let A_i be one of the sets $S_1, \ldots, S_n$ which maximizes $|A_i \cap U|$;
  $U = U \backslash A_i$;
end

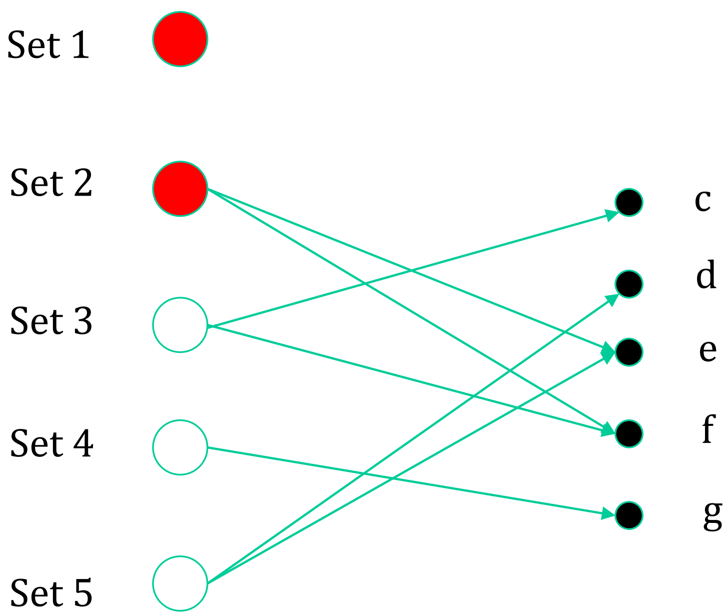


$K = 3$
Selected: Set 1, Covered Elements={a, b, h}

An Example

- The basic idea is to choose the set in each step which contains most of the uncovered elements

Input: V (set of all elements); $S_1, \ldots, S_n; K$
Output: Approximate solution $A_1, \cdots, A_K$ $U = V$
for $i = 1, \ldots, K$ do
  Let A_i be one of the sets $S_1, \ldots, S_n$ which maximizes $|A_i \cap U|$;
  $U = U \backslash A_i$;
end

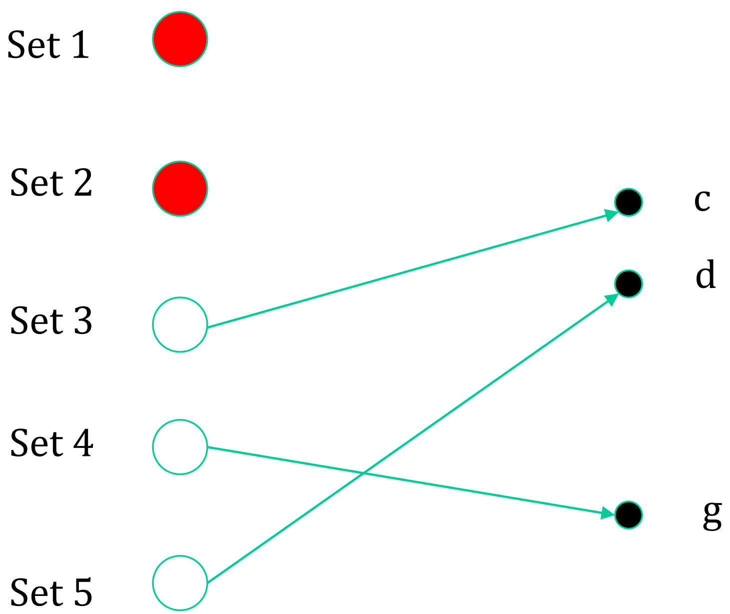


$K = 3$
Selected: Set 1, 2, Covered Elements={a, b, h, e, f}

An Example

- The basic idea is to choose the set in each step which contains most of the uncovered elements

Input: V (set of all elements); $S_1, \ldots, S_n; K$
Output: Approximate solution $A_1, \cdots, A_K$ $U = V$
for $i = 1, \ldots, K$ do
    Let A_i be one of the sets $S_1, \ldots, S_n$ which maximizes $|A_i \cap U|$;
    $U = U \backslash A_i$;
end

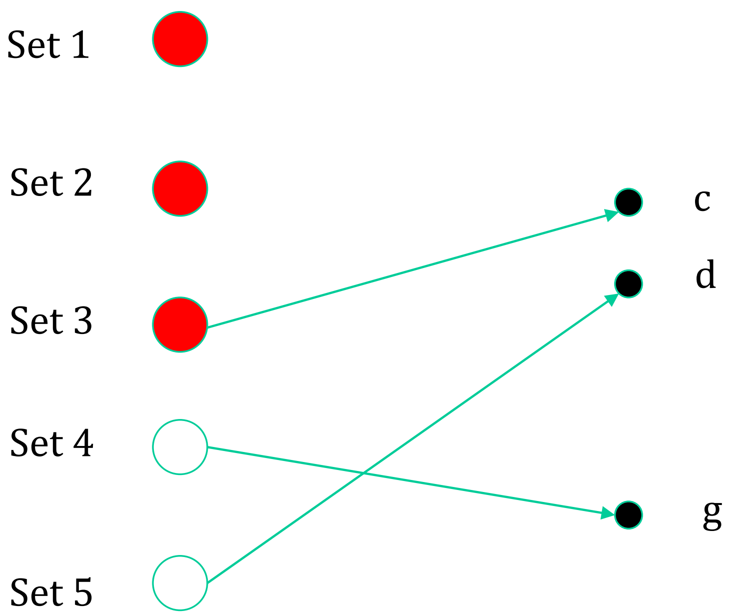


$K = 3$
Selected: Set 1, 2, Covered Elements={a, b, h, e, f}

An Example

- The basic idea is to choose the set in each step which contains most of the uncovered elements

Input: V (set of all elements); $S_1, \ldots, S_n; K$
Output: Approximate solution $A_1, \cdots, A_K$ $U = V$
for $i = 1, \ldots, K$ do
  Let A_i be one of the sets $S_1, \ldots, S_n$ which maximizes $|A_i \cap U|$;
  $U = U \backslash A_i$;
end



$K = 3$
Selected: Set 1, 2, 3, Covered Elements={a, b, h, e, f, c}

Greedy gives (1-1/e)-approx

增加量 after i before i 轮

Lemma 1: For all $i = 1, \ldots, K$, $|A_i \cap U| = |C_i| - |C_{i-1}| \geq \frac{OPT - |C_{i-1}|}{K}$.

Proof: The number of elements covered in the optimal solution but not in the algorithm at the start of iteration i is $\geq OPT - |C_{i-1}|$. Let sets in the optimal solution be $S_1^*, \ldots, S_K^*$. Let $U = V \backslash C_{i-1}$ Obviously,

if all elements in $|C_{i-1}|$ are in OPT, 取等号

U 跟踪未cover的

$$\bigcup_{i=1}^{K} (S_i^* \cap U) = \left(\bigcup S_i^*\right) \backslash C_{i-1}.$$

This implies that

$$\sum_{i=1}^{K} \left| S_i^* \bigcap U \right| \geq \left| \bigcup_{i=1}^{K} \left(S_i^* \bigcap U \right) \right| \geq OPT - |C_{i-1}|,$$

which further implies

$$\max_{i=1,\ldots,K} \left| S_i^* \bigcap U \right| \geq \frac{OPT - |C_{i-1}|}{K}.$$

By definition, $|A_i \bigcap U| \geq \max_i |S_i^* \bigcap U|$ and we are done.

Continued

Lemma 1: For all $i = 1, \ldots, K$, $|A_i \cap U| = |C_i| - |C_{i-1}| \geq \frac{OPT - |C_{i-1}|}{K}$.

Lemma 2: $|C_i| \geq \frac{OPT}{K} \sum_{j=0}^{i-1} (1 - 1/K)^j$ for all $i = 1, \ldots, K$.

Proof: Prove by induction. The base case $i = 1$ is trivial as the first choice $A_1 = C_1$ has at least OPT/K elements by Lemma 1.

For the inductive step, suppose i holds, and we want to prove that it holds for $i + 1$:

归纳

再代值即可

$$\begin{aligned}
|C_{i+1}| &\geq |C_i| + \frac{OPT - |C_i|}{K} \\
&= \frac{OPT}{K} + (1 - 1/K)\,|C_i| \\
&\geq \frac{OPT}{K} \sum_{j=0}^{i} (1 - 1/K)^j.
\end{aligned}$$

The first inequality is by Lemma 1, and the last inequality is from inductive hypothesis.

Finally...

Lemma 2: $|C_i| \geq \frac{OPT}{K} \sum_{j=0}^{i-1}(1-1/K)^j$ for all $i = 1, \ldots, K$.

Proof of Theorem:

$$\begin{aligned}
|C_K| &\geq \frac{OPT}{K} \sum_{j=0}^{K-1} (1-1/K)^j \\
&= \frac{OPT}{K} \frac{1-(1-1/K)^K}{1-(1-1/K)} \\
&= OPT \boxed{\left(1-(1-1/K)^K\right)} \\
&\geq (1-1/e)OPT,
\end{aligned}$$

where the first inequality is from Lemma 2, and the last inequality is from the fact that

$$(1-1/K)^K \leq (e^{-1/K})^K = 1/e,$$

which is obtained from $1 + x \leq e^x$ for all $x \in R$.

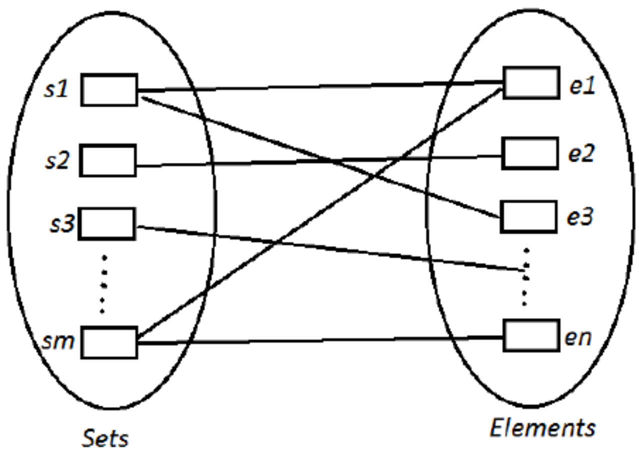
利用 $(1-\frac{1}{K})^K\uparrow$ 的性质

Set Cover Problem

- A universe of elements $V = \{e_1, \ldots, e_n\}$
- A list of (possibly overlapping) sets $\{S_i \subseteq V\}_{i=1}^{m}$

Objective:

We wish to cover all elements with minimum number of sets



Set Cover Problem

- A universe of elements $V = \{e_1, \ldots, e_n\}$
- A list of (possibly overlapping) sets $\{S_i \subseteq V\}_{i=1}^{m}$

Objective: We wish to cover all elements with minimum number of sets

---

Algorithm 1: Greedy Algorithm for Set Cover Problem

Data: A universe $\{e_1, \ldots e_n\}$, a family $S = \{S_1, \ldots S_m\}$.

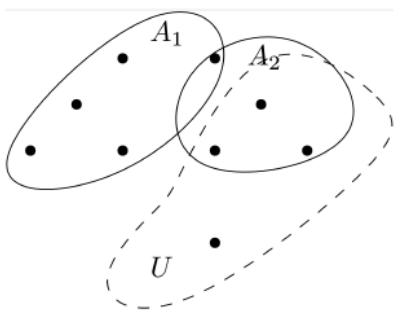
`/*` U `is a set of uncovered elements.` `*/`

$U = \{e_1, \ldots e_n\}$;

while $U \neq \emptyset$, *iteration* i = *1, 2, ... l* **do**

  pick $A_i = \arg\max_{j=1,\ldots m} |S_j \cap U|$

  $U \leftarrow U \setminus A_i$



Analysis

- The max coverage lemma works for any l iterations

lemma 和 max coverage 一样

Lemma: If C_i denotes the set of covered elements at the end of iteration i and C^* denotes the maximum coverage using k sets, then

$$|C_i| \geq \frac{C^*}{k} \sum_{j=0}^{i-1} \left(1 - \frac{1}{k}\right)^j, \quad \forall i = 1, \ldots \ell$$

\# covered elements at ith iteration

≥ lower bound

是一个 geometric sum

Analysis

$n=$ #element

Theorem: The greedy algorithm is $(1+\ln(n))$ -approximation for Set Cover problem.

Proof: Suppose $\boxed{k} = OPT($ set cover $)$. since set cover involves covering all elements, we know that the max-coverage with k sets is $C^* = n$. Our goal is to find the approximation ratio α so that $ALG($ set cover $) = \ell \leq \alpha k$. We apply Lemma at the second last iteration, i.e. $i = \ell - 1$

exp fast

0

$$\begin{cases} |C_{\ell-1}| \leq n-1 \\ |C_{\ell-1}| \geq \frac{n}{k} \sum_{j=0}^{l-2} \left(1-\frac{1}{k}\right)^j = \frac{n}{k} \frac{1-\left(1-\frac{1}{k}\right)^{\ell-1}}{\frac{1}{k}} = n\left(1-\left(1-\frac{1}{k}\right)^{\ell-1}\right) \geq n\left(1-e^{-\frac{\ell-1}{k}}\right) \end{cases}$$

opt # sets to cover all elements

The first inequality is because the uncovered set must contain at least one element, otherwise the algorithm would have stopped before. The second inequality is from Lemma and the fact that $1+x \leq e^x$ for any $x \in (-\infty, \infty)$. From inequalities, we have $ne^{-\frac{\ell-1}{k}} \geq 1$. We can take logarithm on both sides and find the approximation ratio $\alpha \leq 1 + \ln n$ as claimed.

希望 $\frac{1}{k}$ 越小越好,

approximation algrithm
"pre-req poly-time solvable"

$n-1 \geq |C_{i-1}| \geq n\left(1-e^{-\frac{l-1}{k}}\right)$

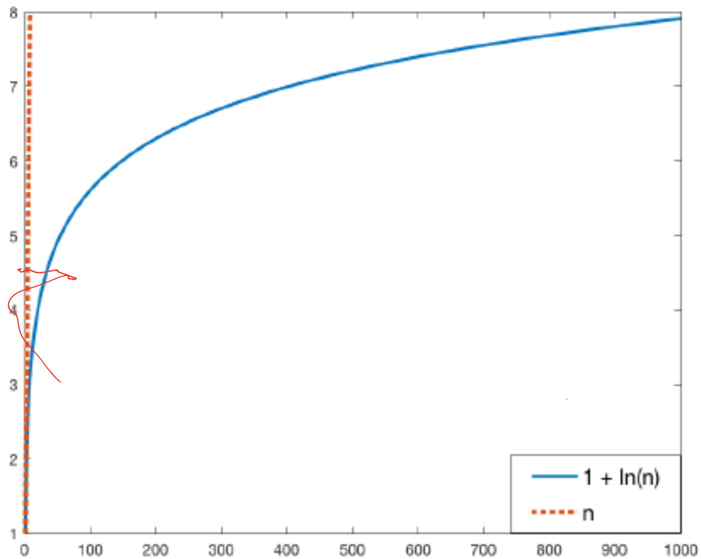
$\Downarrow$

Discussion

$\frac{1}{k} \leq \frac{1}{k} + \ln n \leq 1 + \ln n$

While $\alpha = 1 + \ln(n)$ is not a constant factor, it is still a reasonably good approximation ratio because it grows slowly with the input size n (refer to figure 3). Actually, one cannot get any better approximation algorithm for the Set Cover problem unless $\mathbf{P} = \mathbf{NP}$

$1 + \ln n$



n

Discussion

solve in polynomial time

verify in ...

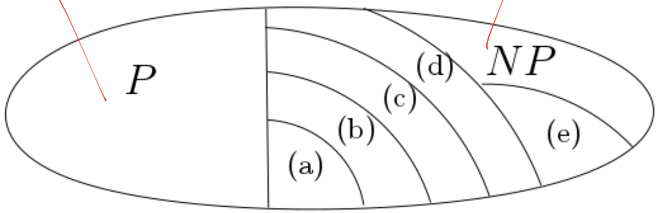


Figure 4: Taxonomy of NP problems (minimization) according to the form of α.

trade-off

(a) $\alpha = 1 + \epsilon$ with running time of $n^{1/\epsilon}, \epsilon > 0$. e.g. PTAS for the Knapsack problem.

(b) α is constant factor, e.g. k-Center Problem, Maximum Coverage, TSP. $\alpha = 2 / 1.5$

$\alpha = 1 - \frac{1}{e}$

(c) $\alpha = 1 + \log(n)$. e.g. Set Cover problem.

(d) $\alpha = 1 + \log^2(n)$. (poly-log) 比(b)难

分5类

(e) α is linear in n. 最[illegible]

若 P ≠ NP, we want to develop "appro. algo"

供应链 基本上都是NP-hard

- Inventory
- Transportation
- matching ⟶ integer variable 大多是NP-hard.

LP是P，可solve fast

3类很好的Algo来处理NP-hard

1. greedy Algo
2. local search
3. LP-relaxation.

e.g. 处理 interger variable $X \in \{0,1\}$

$\Longrightarrow 0 \le X \le 1$

rounding (many ways)

∃integer solution

除了solve之外，还要设计Algo

来verify in polynomial time.

e.g. TSP with edge cost integer

$\Rightarrow$ NP-complete (verify in poly time)

TSP with fractional edge cost

(cannot even verify in poly-time)

most effect $\underline{\leq \alpha \, OPT}$ (而并非找OPT)