2019 Lec 8

2019年5月8日 11:13

Knapsack problem

① Greedy.

$$\frac{\text{profit}(a_1)}{\text{size}(a_1)} \geq \frac{\text{profit}(a_2)}{\text{size}(a_2)} \geq \cdots \cdots \geq \frac{\text{profit}(a_n)}{\text{size}(a_n)}.$$

Greedy solution: $\{a_1, a_2 \cdots a_k\}$. ← arbitrary bad.

improvement: $\{a_1, \cdots a_k\}$ OR $\{a_{k+1}\}$

---

② Dynamic programming.

$f(i, y)$: 前i个物品中 选一个子集S. s.t. $\sum_{i \in S} \text{profit}(a_i) = y$.

$$= \min_{S \subseteq \{1,2,\cdots i\}} \left\{ \sum_{i \in S} \text{size}(a_i) \;\middle|\; \sum_{i \in S} \text{profit}(a_i) = y \right\}.$$

case 1. $i \notin S$. case 2 $i \in S$.

$$f(i, y) = \min\left(f(i-1, y),\ f(i-1, y - \text{profit}(a_i)) \right).$$

$f(1, y)$ $f(1, 0) = 0$.

$f(1, \text{profit}(a_1)) = \text{size}(a_1)$.

$f(n, y) \leq B$.

Time: $n \times \sum_{i=1}^{n} \text{profit}(a_i)$ $W = \max_i \text{profit}(a_i)$

$O(n \times n \times W)$ $n \cdot \log W$,

K. $\text{profit}'(a_i) = \left\lfloor \frac{\text{profit}(a_i)}{K} \right\rfloor \times K$

→ $(\text{profit}'(\cdot), \text{size}(\cdot)) \Leftarrow$ DP. Time $O(n^2 \times \lfloor \frac{W}{K} \rfloor)$

$(\text{profit}'(\cdot), \text{size}(\cdot)) \Leftarrow$ DP $\to$ set S.   Time $O(n^2 \times \lfloor \frac{W}{K} \rfloor)$

1° S is a feasible solution.

2° $\text{Algo} = \text{profit}(S)$ V.S. $\text{profit}(S^{OPT})$

$\forall$ ~~object~~ a_i. $\text{profit}'(a_i) + K \geq \text{profit}(a_i) \geq \text{profit}'(a_i)$

$\text{Algo} = \text{profit}(S) \geq \text{profit}'(S) \geq \text{profit}'(S^{OPT}) \geq \underbrace{\text{profit}(S^{OPT})}_{OPT} - K \cdot |S^{OPT}|$

S: $(\text{profit}'(\cdot), \text{size}(\cdot))$ 的最优解.

$\geq OPT - K \cdot n$ Time: $O(n^2 \cdot \frac{W}{K})$.
W

$K \triangleq \frac{W}{n} \cdot \varepsilon$ $W \leq OPT$ Time: $O(\frac{n^3}{\varepsilon})$ Space: $O(\frac{n^2}{\varepsilon})$

$\text{Algo} \geq OPT - W \cdot \varepsilon \geq OPT(1-\varepsilon)$

Approximation ratio : $1-\varepsilon$

FPTAS: $1-\varepsilon / 1+\varepsilon \Leftarrow$ Approximation ratio.

Time: $\text{poly}(n, 1/\varepsilon)$. $n^3 \times \frac{1}{\varepsilon}$

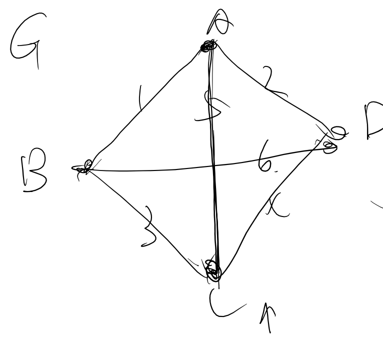
PTAS: $1-\varepsilon / 1+\varepsilon \longleftarrow$ Approximation ratio

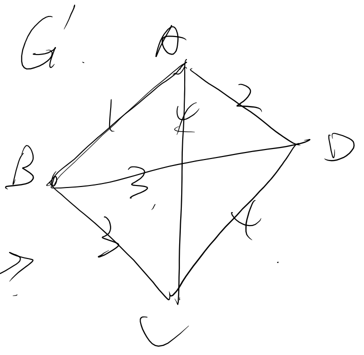
Time: if ε is constant, $\text{poly}(n)$

$O(n^{\frac{1}{\varepsilon}})$ $O(n^{2^{\frac{1}{\varepsilon}}})$

一个

steiner Tree.



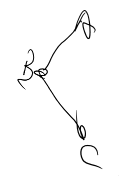


R: A, C.

Tree T': {A C}

↓

Tree T:



Thm: Algo A: metric steiner tree problem α-approx.

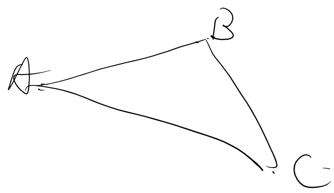
⇓

Algo A': steiner tree problem α-approx.

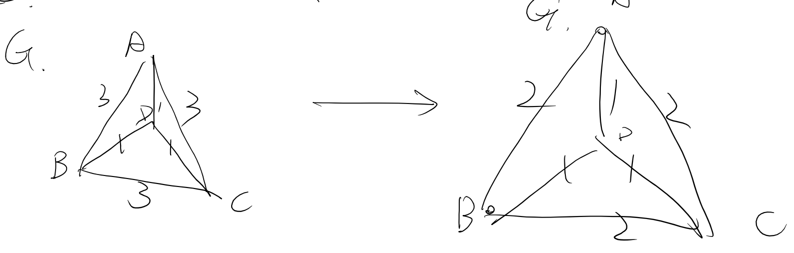
$$
\begin{tikzcd}
\text{ST instance } G=(V,E) \arrow[rr, "①"] \arrow[d, squiggly, "③\ \alpha\text{-近似}"'] & & \text{metric ST instance } G'=(V',E') \arrow[d, "A\quad \alpha\text{ 近似}"] \\
\text{Tree } T & & \text{Tree } T' \arrow[ll, "②"]
\end{tikzcd}
$$

① $G=(V,E) \longrightarrow G'(V,E')$

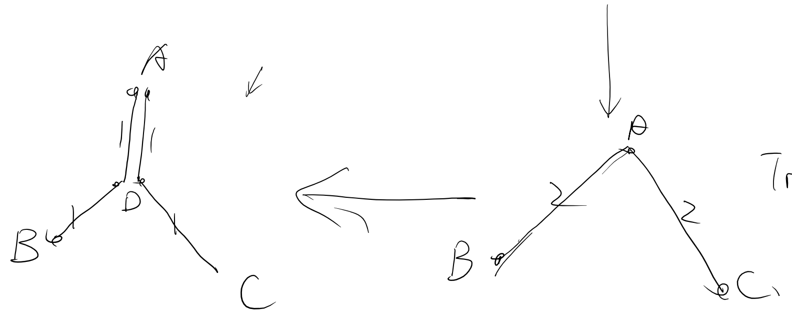
weight → shortest path length.



② Tree T'.



A



Tree T'.

Remove multi-edge cycle.

$$w(T') \le \alpha \cdot OPT(G')$$

$$\Downarrow$$

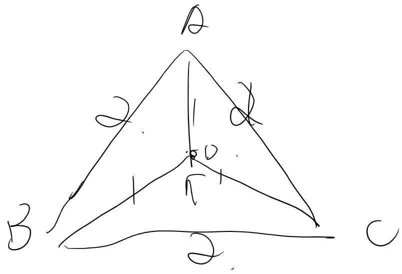
$$w(T) \le \alpha \cdot OPT(G)$$

$$w(T) \le w(T')$$

$$OPT(G) \ge OPT(G')$$

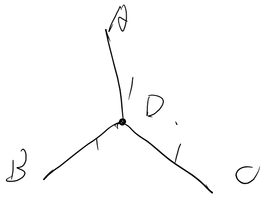
$V(R,S)$ E $\quad$ $V(R,S)$ E'

Approximation Algorithm.

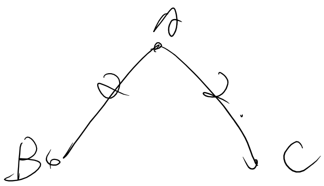


$R: \{A, B, C\}$.

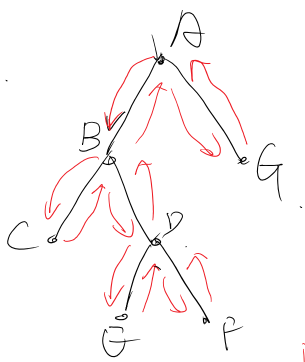
$OPT = 3$.



$Algo = 4$



OPT tree.



PATH: A B C B D E D F D B A G A.

$w(PATH) = 2 \cdot OPT$

PATH': $A \to B \to C \to D \to E \to F \to G \to A$.

PATH': $A \to B \to C \to D \to E \to F \to G \to A$.

Required: $\{A, C, E, F, G\}$.

$w(PATH') \le w(PATH)$

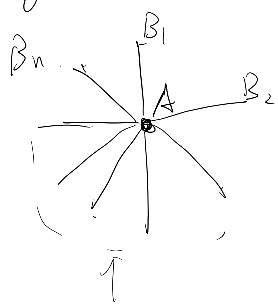
PATH'': $A \to C \to E \to F \to G \to A$.

$w(PATH'') \le w(PATH')$

$\{A, C, E, F, G\}$ spanning tree. $A \to C \to E \to F \to G$. ↙ PATH'''

$$Algo \le w(PATH''') \le 2 \cdot OPT$$

Tight example.

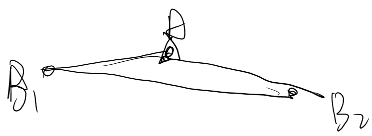


$w(A, B_i) = 1$ $\quad$ $OPT = n$.

$w(B_i, B_j) = 2$. $\quad$ $Algo = 2(n-1)$

$R = \{B_i\}$.

$$\sim \frac{2(n-1)}{n} \sim 2 - \frac{2}{n}.$$



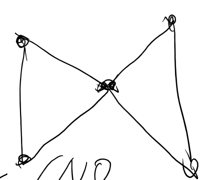
A: $(B_1, B_2) \leftarrow$

(B_1, B_3)

(B_2, B_4)

General TSP.

Thm: Assume $\exists$ Algo A: approx $\alpha(n)$, then we can solve Hamilton cycle problem.



HC instance $G=(V,E)$

$\downarrow$

TSP instance $G'=(V', E')$ $\xrightarrow{A}$ cycle.

YES / NO.

TSP instance $G=(V,E)$. ——— cycle.

$G' = (V, E')$ $\quad w(i,j) = \begin{cases} 1 & (i,j)\in E \\ \alpha(n)\cdot n & (i,j)\notin E \end{cases}$

HC $\quad$ G: has a Hamilton cycle $\Rightarrow$ TSP $OPT = n$.

G: ~~has not~~ a HC $\not\Rightarrow$ TSP $OPT \geq \alpha(n)\cdot n + 1$

1°. $Algo(A) \leq n\cdot\alpha(n) \Rightarrow$

2°. $Algo(A) \geq n\cdot\alpha(n)+1 \Rightarrow OPT > n$

Assume $OPT = n$. $\quad Algo(A) \leq n\cdot\alpha(n)$

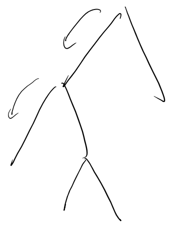
---

metric TSP.

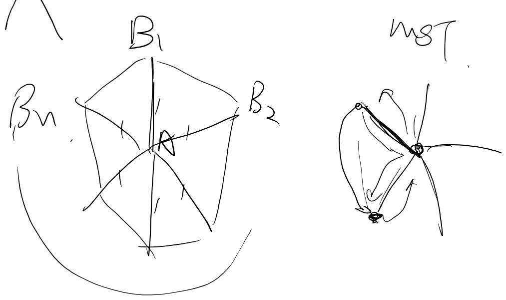
mini spanning tree

$w(PATH) \leq 2\cdot w(Tree) \leq 2\cdot OPT$

OPT



tight example.



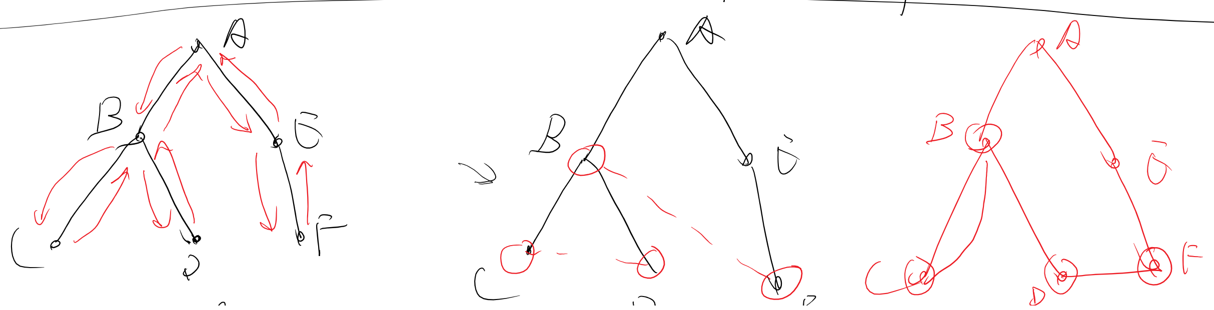
TSP tour. $A\ B_1\ B_2\ B_3$ ——— $B_n - A$.

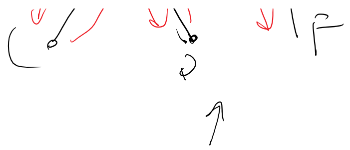
$B_i\, B_{i+1} = 2$.

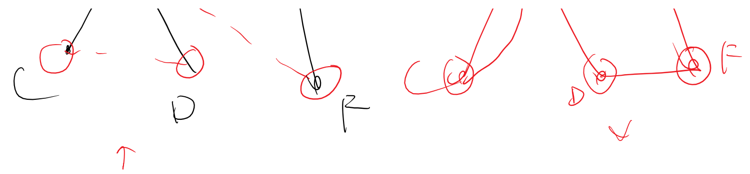
$Algo = 2(n-1)+2 = 2n$.

$OPT \quad A - B_1 - B_3 - B_5 \cdots - B_n - \cdots - A$

$= n+1$





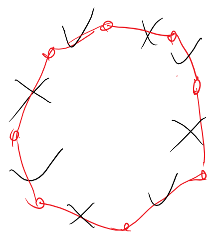


A B C D B F E A   A B C B D F E A.

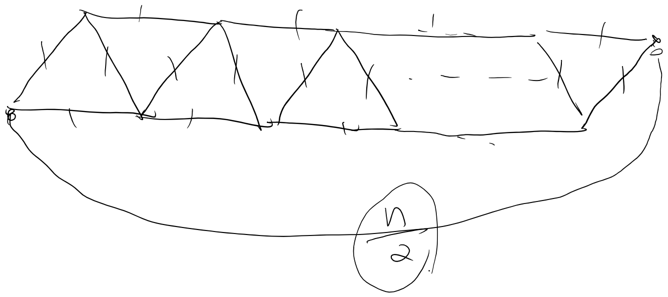
$$\text{Algo}: \le \underbrace{w(\text{Tree})}_{\le OPT} + \underbrace{w(\text{min weight perfect matching})}_{\le \frac{OPT}{2}} \le 1.5\, OPT$$

$$w(\text{min weighted perfect matching}) \le w(\text{min weight maximum matching}) \le \frac{OPT}{2}$$

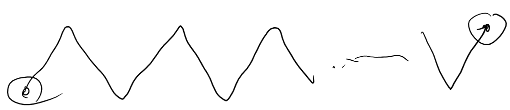
OPT



tight example

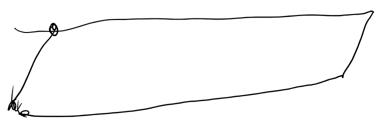


min spanning tree



$\text{Algo}: n-1+\frac{n}{2}$

OPT



$OPT = n$