110 minute
- 1 yes/no
- 2 max flow alg

20 min

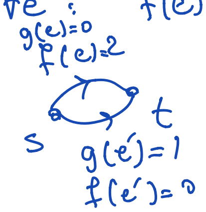
do
- 3
- 4
- 5
- 6

22.5 minut

P1)

a) f, g two diff flows on (G, s, t) $v(f) \geq v(g)$

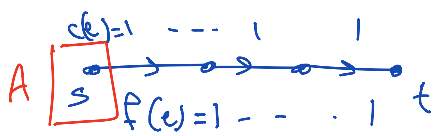
$\Rightarrow \forall e:\ f(e) \geq g(e)$. False

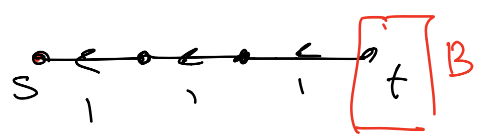


---

b) f is a max flow B is set of vertices that can reach t in G_f. $(V-B, B)$ is a min-cut.

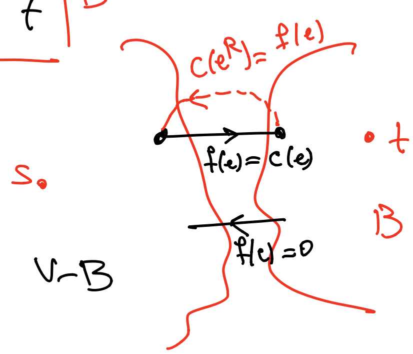
True.





$$v(f) = \sum_{\substack{e \text{ out of} \\ V-B}} f(e) - \sum_{\substack{e \text{ into} \\ V-B}} f(e)$$

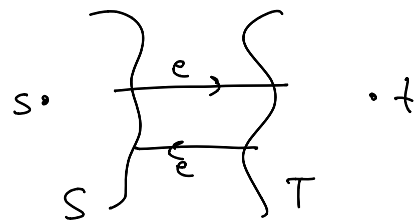
$$= \sum_{\substack{e \text{ out of} \\ V-B}} c(e) - \sum_{\substack{e \text{ into} \\ V-B}} 0$$



$= \text{cap}(V-B, B)$

---

c) f is max-flow (S,T) mincut. $\forall e$ s.t. endpoints of e on diff sides of the cut we have $f(e) = c(e)$. False



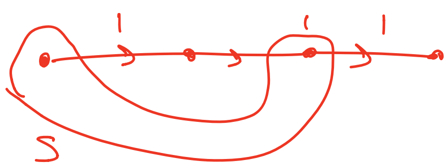
$v(f) = \sum_{e \text{ out of } S} f(e) - \sum_{e \text{ into } S} f(e)$

an equality $\leq \sum_{e \text{ out of } S} c(e) - \sum_{e \text{ into } S} 0$

$= \text{Cap}(S,T)$

So $f(e) = c(e)$ $\forall e$ out of S

$f(e) = 0$ $\forall e$ into S.



---

d) B is NP-hard & $A \leq_p B \Rightarrow A$ is NP-hard.

False

e) A is NP-hard & $A \leq_p B \Rightarrow B$ is NP-hard.

Recall: A is NP-hard iff $\forall C \in NP$ $C \leq_p A$.

$C \leq_p A \leq_p B \Rightarrow C \leq_p B$ True

---

f): If $P \neq NP \Rightarrow$ every problem in NP require exp time

False.

$P \subseteq NP$ and can be solved in poly-time.

g) $A \in P \Rightarrow A \le_p B \quad \forall B \in NP$.

True. Can just solve A in poly time.

If $A \le_p B$ then

h) G is weighted n vertices, m edges no neg cycle.

BF will be stabilized in $\le n-1$ iterations, True

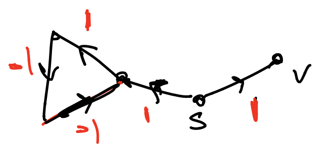
$Opt[v, i]$ = shortest path to v using $\le i$ edges.

$\forall\ i \ge n-1$, $Opt[v,i] = Opt[v, i+1]$.

i) G n ver m edges conta neg cycle.

shortest path with n edges < shortest path with $n-1$ edges.

False



j) True

$T(n) = 24\, T(n/3) + O(n^2)$.

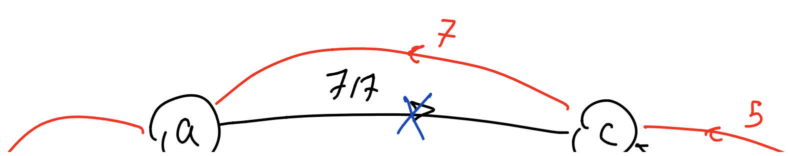
$a = 24 \quad b = 3 \qquad n^2 \text{ vs } n^{\log_3 24}$

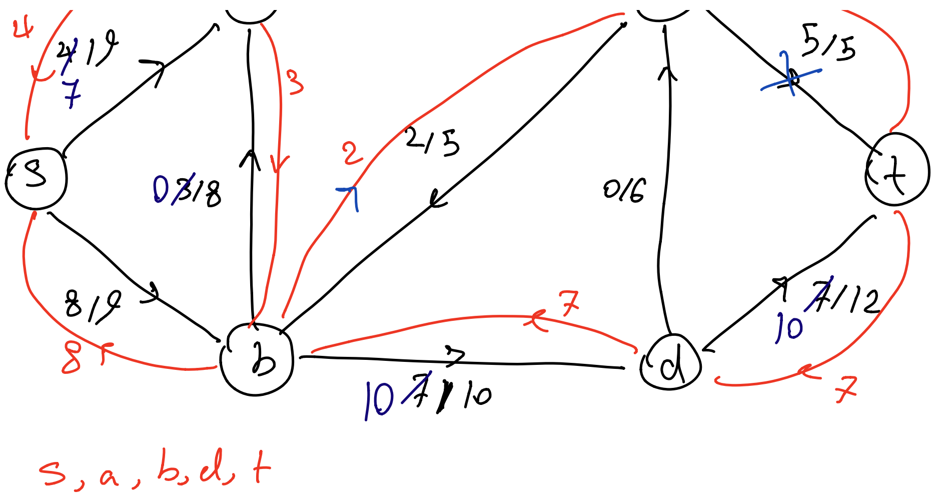
k) $T(n) = T(9n/10) + T(3n/20) + O(n)$.

False

if $3n/10$ was $n/20$ then lin time

2)





---

3) a) OPT (n), OPT (1) = 1

$$OPT(i) = \min_{1 \le j \le i} \{ OPT(j)/j + w(j) \}$$

~~4a) ba)~~

4) $x_1 \dots x_n \in \{0,1\}$

Group-Test $(i,j) = x_i \vee \dots \vee x_j$.

To output ~~x~~ i when $x_i = 1$ if it exists.

Each time $[l, r]$ ~~item~~ interval.

Def mid $= \frac{l+r}{2}$.

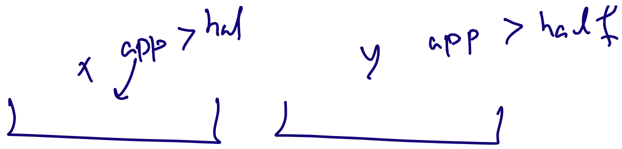
if Group-Test $(l, mid) = 1$

$r = mid$

else

$l = mid$.

$b_1 \dots b_n$ ~~IS~~ there a numb that app $> n/2$ times.



$T(n) = 2T(n/2) + O(n)$

So $T(n) = O(n \lg n)$

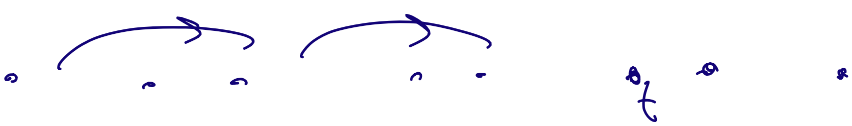
main idea: If x is dominant in $[1, -, n]$
it is either dominant in $[1, -, n/2]$
or domine in $[n/2+1, -, n]$.

6d e.g. Given a DAG count # paths ending at v.

A path ending at u is either just of length 0
or there is a vertex before u.

$$OPT(u) = 1 + \sum_{w : w \to u} OPT(w).$$

6) Count # paths from v to t, for a fixed given t.



For evy vertex v bef t.

$$OPT(v) = \sum_{u : v \to u} OPT(w.$$

$$OPT(t) = 1.$$

Remember to fill it up downwards.

---

7) Number Partition Problem.

Given $y_1 \sim y_n$ Is there a set S. s.t.

$$\sum_{i \in S} y_i = \sum_{i \notin S} y_i$$

a) Num Par Prob is in NP.

Given the set S certifier sums up numbers in S and numbers in $\bar{S}$ and see if they are the same. of course runs in poly time

b) Subset Sum Problem. Is there S s.t.

$$\sum_{i \in S} y_i = 1?$$ (the number can be any)

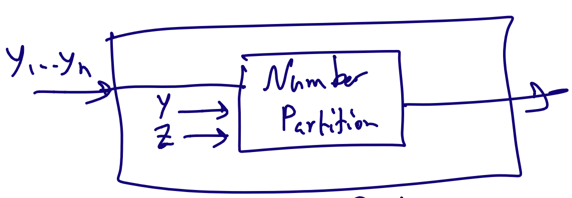
Q: Subset Sum $\leq_P$ Number Partition.

y_1 -- y_n

Supp $\exists A$ s.t. $\sum_{i \in A} y_i = 1$ $\quad \sum_{i \notin A} y_i = \text{Sum} - 1$

Def: $\text{Sum} = \sum_i y_i$ $\qquad y = M + \text{Sum} - 2 \qquad z = M$

$M >> \text{Sum}$



Subset Sum

Supp yes instance of subset sum.

So $\exists A$ s.t. $\sum_{i \in A} y_i = 1$ $\quad \{A \cup y\} \quad \{\bar{A} \cup z\}$

$$y + \sum_{i \in A} y_i = M + \text{Sum} - 1 = z + \sum_{i \notin A} y_i .$$

yes inst of Number Partition

Conversely Supp Num Partition $(y_1, \ldots, y_n, y, z) = $ yes.

y and z must be mapped to diff sets.

So, supp $\exists A \subseteq \{1, \ldots, n\}$ s.t

$$y + \sum_{i \in A} y_i = z + \sum_{i \notin A} y_i$$

$$\Rightarrow \cancel{M} + Sum - 2 + \sum_{i \in A} y_i = \cancel{M} + \sum_{i \notin A} y_i$$

$$\Rightarrow \sum_{i \in A} y_i = \sum_{i \notin A} y_i - Sum + 2.$$

$$= \sum_{i \notin A} y_i - \sum_i y_i + 2$$

$$= -\sum_{i \in A} y_i + 2 \Rightarrow \cancel{2} \sum_{i \in A} y_i = \cancel{2} \; 1$$

8) $(s_1, f_1) \ldots (s_n, f_n)$

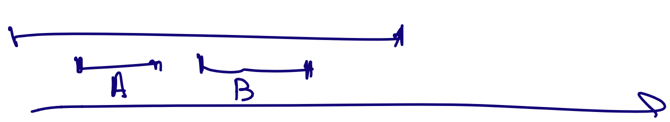
$f_1 \leq \cdots \leq f_n.$

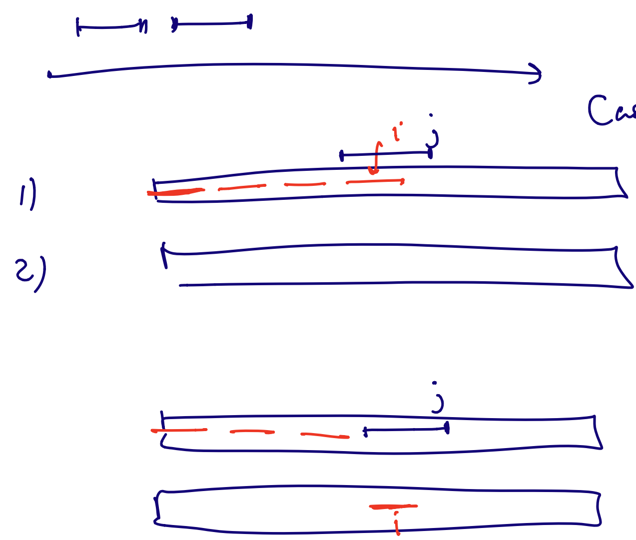
while there is a req (s_i, f_i) comp with either A_1 or A_2

add the first unused req to A_1

add the first unused req to A_2

$(1, 2)$





Case1

switch j with i

OPT remain val

BC $f_j \geq f_i$