4 problems

1. Yes/No problem No explanation is needed.

2, 3, 4 Algorithm deisg /proof.

connect (it will be in the problem)

if not said it may be disconnected.

60 min 15 minutes per problems

Rubric if your ALG/ide is correct (10)

20 you get ≥ half.

→ major detal −5, −6

→ minor del −1, −2.

Spend at lest 10 min per problem.

Read statement carefully.

Should you have questions? Send email to staff list

take exam 9:30 − 2:40 response

Sample Midterm:

P1) a) $n^{2.1} = O(n^2 \lg n)$ False

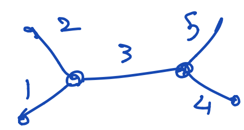
$n^{2.1} \leq \Theta\, n^2 \lg n$

b) There is a poly time ALG to decide whether a graph is bipartite ✓

use BFS.

c) an undirected connected graph has a unique heaviest edge e, the e is not in any MST

False

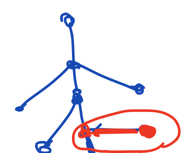


d) If all edge have weight 1, then there is $O(m+n)$ to find MST. ✓
B/C any spanning is a MST.

e) $T(n) \leq \underbrace{10}_{a} T(\underbrace{n/3}_{b}) + n^{\overbrace{3}^{k}} \Rightarrow T(n) = O(n^3)$. ✓

$a < b^k$

2) Given a tree T, with 2n vertices. Find a perfect matching poly-time ALG.



P(n) = ~~Ale~~ For any acyclic ~~tree~~ graph with 2n nodes, ALG outputs a PM iff it exists.

P(1) = • • no edge then no PM
•—• a PM exists

IH: Assume P(n-1).
IS: Prove P(n). Let G be an arbitrary acyclic graph. with 2n vertices. Every connected comp of G is a tree.
If G has a vertex v of $d(v)=0$ then output no PM.
O.W. G has a leaf v (B/C every tree comp has a leaf)
Let u be neighbour of v.

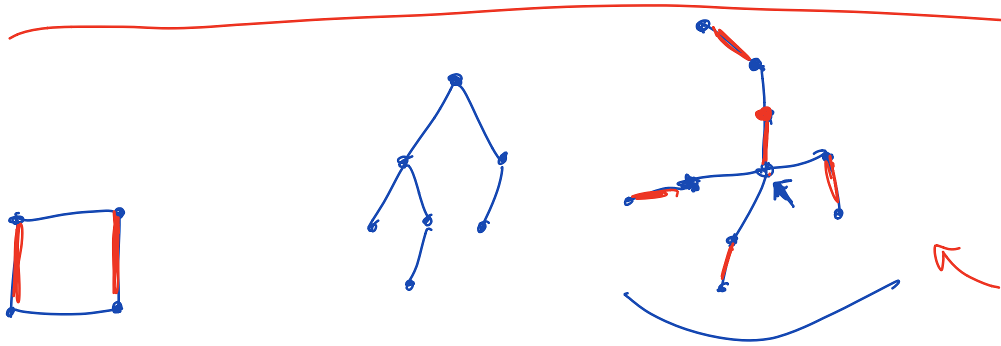
$\boxed{\text{Then in evry PM matching of } G}$ v is matched to u.

Match v to u, and $G' = G - u - v$. G' is acyclic
has $2(n-1)$ vertices

By IH we find a PM in G' iff it exist.

If G' has a PM, add (u,v) and we find a PM in G.

If G has a PM M, then $(u,v) \in M$, so $M - (u,v)$ is a PM of G' and my algorithm correctly outputs a PM.



3) a_1 ___ a_n

4) Given a sorted array $A[1] < \dots < A[n]$
Is there i s.t. $A[i] = i$.

ALG: find (l, r)

If $(l = r)$
then check if $A[l] = l$ output l, else output NO.

$m = (l+r)/2$.

If $A[m] = m$

output m $\quad$ // $A[i] > i$ for all $i > m$

else if $A[m] > m$

find (l, m)

else

find (m, r)

Pf. Since A is sorted and distinct, $A[i+1] \geq A[i]+1$ for all i.

Fact: If $A[m] > m$ then for all $i > 0$

Pf $A[m+i] \underset{\substack{\text{sorted seq} \\ \text{distiint}}}{\geq} A[m]+i > m+i$ $\quad$ no solution right of m

Similarly if $A[m] < m$, then for all $i > 0$

$A[m-i] \leq A[m] - i < m-i$ $\quad$ no solution left of m

So this justifies the recursion if $A[m] > m$ the solution can only be in $[l, m]$ interval and if $A[m] < m$ it can only be in $[m, r]$ interval.

HW 4 - P4.

$$a_1 \quad \underbrace{a_{i-1} < a_i > a_{i+1}}_{\text{maximal el}} \quad a_n$$

$P(k) =$ Given a seq $a_l \cdots a_r$ s.t. $r - l = k$ my ALG finds a maximal element.

$m = \frac{(l+r)}{2}$ mid point

Base Case $P(1)$ just output single element

IH: For some $k > 1$, Assume $P(j)$, for all $1 \le j \le k-1$.

IS: Prove $P(k)$. Assume $[l, r]$ arbitrary is given s.t. $r - l = k$

$m = (l+r)/2$.

Case 1) $a_{m-1} < a_m > a_{m+1}$ just output a_m.

Case 2) $a_{m-1} > a_m$. Since $m - l < k$, By $P(m-l)$ on $[l, m]$ finds a local max.

$[\boxed{a_l \cdot a_m} \quad a_r]$

For a_m being a local max in $P(m-l)$ we just need $a_m > a_{m-1}$ but for it to be a local maxim in $P(k)$ we need $a_{m-1} < a_m > a_{m+1}$

$a_m < a_{m-1}$ So a_m is not a local maxim of $[l, m]$ interval and $P(m-l)$ does not return a_m.

any other number that $P(m-l)$ returns is also a local max of $P(k)$.

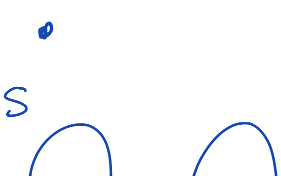
Case 3) O.w. $a_m < a_{m+1}$ similarly return output of $P(r-m)$ on interval (m, r).

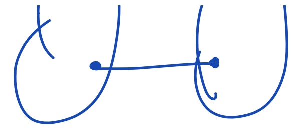
Output is not a_m B/C by $a_m < a_{m+1}$, a_m is not a local max of $P(r-m)$

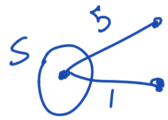
Any other local max is also a local max of $P(k)$.

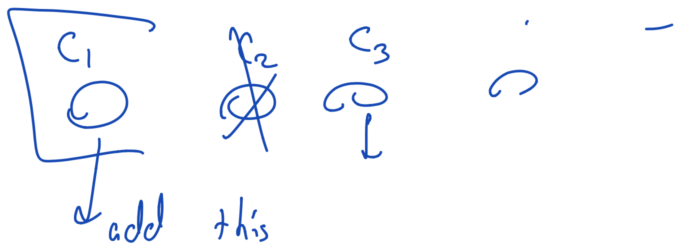
Cut Prop

S









If C_1 not in OPT