4 problems
1. Yes/No problem. No explanation is needed.
2, 3, 4 Algorithm design / proof.
    connect (it will be in the problem)
    if not said it may be disconnected.
60 min 15 minutes per problem
Rubric if your ALG/ide is correct 10
20 points
you get 7 right.
major 6 deducted -5, -6
minor deducted -1, -2.
Spend at least 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:
1) a) \( n^{2.9} = O(n^{2.5}) \) False
   \( n^{3.1} \leq \Theta n^{2.5} \)
   b) There is a poly time ALG to decide whether a graph
   is bipartite \( \checkmark \)
   use BFS.
   c) an undirected connected graph has a unique
   heaviest edge \( e \), the \( e \) is not in any MST
   False
1) If all edge have weight 1, then there is $O(mn)$ to find MST.✓
   B/C any spanning is a MST.

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

   $P(n)$ = Ask forget any tree with $2n$ nodes, ALG outputs
           a PM iff it exists.

   1. $P(1) = \bullet \bullet$ 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. Evy connected comp of $G$ is a tree.
       If $G$ has a vertex $v$ of $d(v)=0$ then output no PM.
       P.W. $G$ has a leaf $v$ (B/C evy tree comp has a leaf)
       Let $u$ be neighbour of $v$. 
Then in any PM matching of $G$, $v$ is matched to $u$.

Match $v$ to $u$, and $G' = G - u - v$. $G'$ is cyclic if it has $2m - 1$ vertices.

By IH, we find a PM in $G'$ iff it exists.

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) $a_1 \leq a_n$

4) Given a sorted array $A \subseteq \{1, \ldots, n\}$

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 $\phi$.

$m = (l + r) / 2$

If $A[m] = m$
Given a seq \( a_l \), output an

\[ P(k) = \text{Given a seq } a_l, a_r \text{ s.t. } r - l = k \]

my ALG finds a maximal element.

\[ m = \frac{(l + r)}{2} \text{ mid point} \]

Base Case: \( P(1) \) just output single element.
IH: For some \( k \geq 1 \), Assume \( P(j) \), for all \( 1 \leq j \leq k - 1 \).

IS: A proof \( P(k) \). Assume \( \exists (l, r) \) arbitrary is given s.t. \( nL = k \)
\[ m = \frac{(l+r)}{2} \]

Case 1) \( a_{m-1} < a_m \geq a_{m+1} \) just output \( a_m \).

Case 2) \( a_{m-1} \geq a_m \). Since \( m - l < l \), By \( P(m-l) \) on \( (l, m] \)
finds a local max. \( a_L \cdot a_m \geq a_J \)

For \( a_m \) being a local max in \( P(m-l) \) we just need \( a_m > a_{m-1} \) but for \( a_m \) to be a local max 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 max in \( (l, m] \)
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) \( P(m, a_m < a_{m+1} \) simply return output
of \( P(m-r) \) 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 \( P(m-r) \)
Any other local max is also a local max
of \( P(k) \).

Cut Prop S
If \( c_i \) not in \( \text{OPT} \)

\[ c_i \rightarrow \text{OPT} \triangleleft \text{improve \ OPT} \]

in match