1. You may assume that the base case (for $n_0 < n$) takes constant time. Even if you had any other assumptions, all those were considered for grading.

(a) (5 pts) $T(n) = 9\,T(n/3) + n^2$
Using Master's method. $f(n) = n^2 = \theta(n^{\log_3 9} \lg^0 n)$ is the case 2, implies $T(n) = \theta(n^2 \log n)$

(b) (5 pts) $T(n) = T(n - 1) + n^2$
We can use substitution method for this. Suppose $T(k) \le ck^3$ for all $k < n$. Then $T(n) = T(n-1) + n^2 \le c(n-1)^3 + n^2 = cn^3 - \big((3c-1)n - 3c\big)n - c < cn^3$ where $c > 1$ and $n > 2$. Similarly, if we assume $T(k) \ge dk^3$ for all $k < n$, then $T(n) = T(n-1) + n^2 \ge d(n-1)^3 + n^2 = dn^3 + \big((1-3d)n + 3d\big)n - d > dn^3$ where $d > \frac{1}{3}$ and $n > 2$. Therefore we can find constants d and c satisfying $dn^3 \le T(n) \le cn^3$ for $n > 2$, by definition, $T(n) = \theta(n^3)$.

(c) (5 pts) $T(n) = 2\,T(n/4) + \sqrt{n}$
Using Master's method. $f(n) = n^{1/2} = \theta(n^{\log_4 2} \lg^0 n)$ is the case 2, implies $T(n) = \theta(n^{1/2} \log n)$

(d) (5 pts) $T(n) = 5T(n/5) + n \lg n$
Using Master's method. $f(n) = n \lg n = \theta(n^{\log_5 5} \lg^1 n)$ is the case 2, implies $T(n) = \theta(n \log^2 n)$

Grading policy : For each subproblem, 3pts(logic) + 2pts(answer) basically, -1pt for minor calculation error, and you get total 2pts for half proof(only for upper or lower bound). If there are critical logic errors, you get total 0pt. Any method is acceptable, for example substitution method, master method, recursion tree, unrolling, etc.

2. If you only wrote the answer without proof, no score was given.

(a) True
Let T be an MST of graph G. Given a connected subgraph H of G, show that $T \cap H$ is contained in some MST of H. Suppose not. Then there is an edge $e \in T \cap H$ across some cut $(S|V-S)$ of H such that another edge across the cut, e', is lighter. However, H is a subgraph of G, so e' is lighter than e across the cut $(S|V-S)$ of G. We could thus swap e' for e in T to get a lighter spanning tree, contradicting that T is an MST.

(b) True
A directed graph G is a directed acyclic graph (DAG) if there is no directed cycle in G. For a directed graph G, its meta-graph G^{SCC} is a DAG.

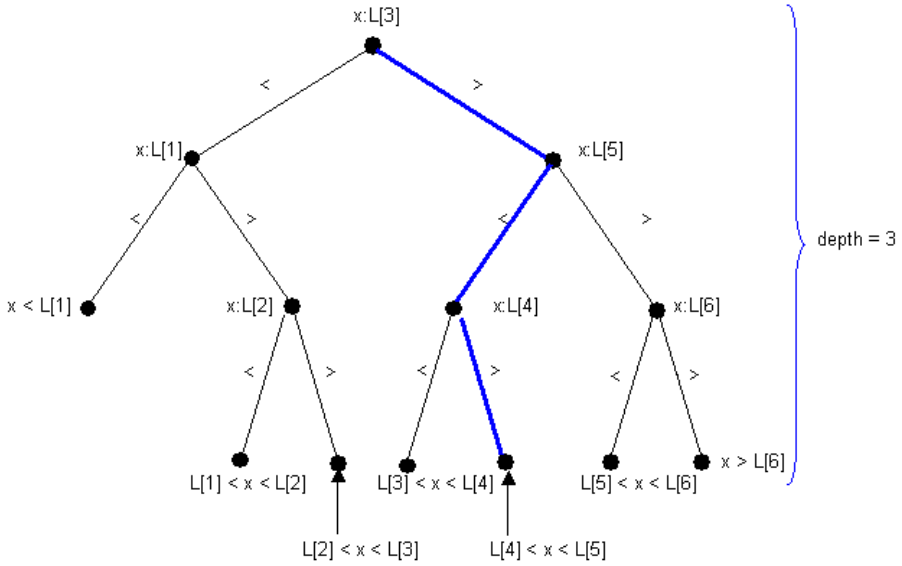
(c) False
MST algorithm does not mean the shortest distance between vertices.

(d) True

A maximum spanning tree is a spanning tree of a weighted graph having maximum weight. It can be computed by negating the weights for each edge and applying Kruskal's algorithm.

3. For a proof, we can make a decision tree with n leaf-nodes and its height h is ceil(logn).Height h would be # of comparison steps and h should be greater than or equal to logn.

e.g., if the list A consists of A[1], A[2], [3], A[4], A[5], A[6] where n is 6,



4. It should satisfy c1(f(n) + g(n)) ≤ max(f(n), g(n)) ≤ c2(f(n) + g(n)) for max(f(n), g(n)) = Θ (f(n) + g(n)).

Lower bound: max(f(n), g(n)) ≥ (f(n) + g(n))/2, note that f(n) ≤ max(f(n), g(n)) and g(n) ≤ max(f(n), g(n)).

Upper bound: max(f(n), g(n)) ≤ (f(n) + g(n)), note that f(n) ≤ f(n) + g(n) and g(n) ≤ f(n) + g(n).
c1 = 1/2, c2 = 1

5 pts for each lower bound and upper bound.

5. Always choose the median as pivot using median of medians algorithm. It takes $O(n)$ time to find the median. Then $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$. Therefore, $T(n) = O(n \log n)$.

5 points for using the median as pivot. 5 points for analyzing the running time.

6. Selection(L,k)
1: Get median m of L by black box.
2: n = len(L)
3: Make a partition P1 and P2 with pivot m. Every elements in P1 are lower than m. Otherwise, they are greater than m.
4: If k = [n/2] and n is odd -> return m
5: If k<[n/2] -> return Selection(P1, k)
6: If k>[n/2] -> return Selection(P2, k - ceil(n/2))

Now let's analyze the complexity of this algorithm. In line1, we use black box, and it takes O(n) times. In line 3, we make partition, and it also takes O(n) times. Note that |P1| and |P2| are both lower than n/2. So, in worst case, we call Selection function again with size at most n/2. It yields following recurrence relation.

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

Using master theorem or any other method to solve recurrence relation, you can get T(n)=O(n).

<Grading Criteria>
Completely wrong algorithm : 0
Valid, but not linear algorithm : 3
Error in analyzing time complexity : 5
Not concerning partition process : -3
Minor error : -1

7. There can be several ways to do this. Run a depth-first search on G. When a back edge is detected for the first time, print or return *true* and exit immediately. Otherwise, print or return *false*. Assume that there is no self-loop and edge $(a, b) = (b, a)$.

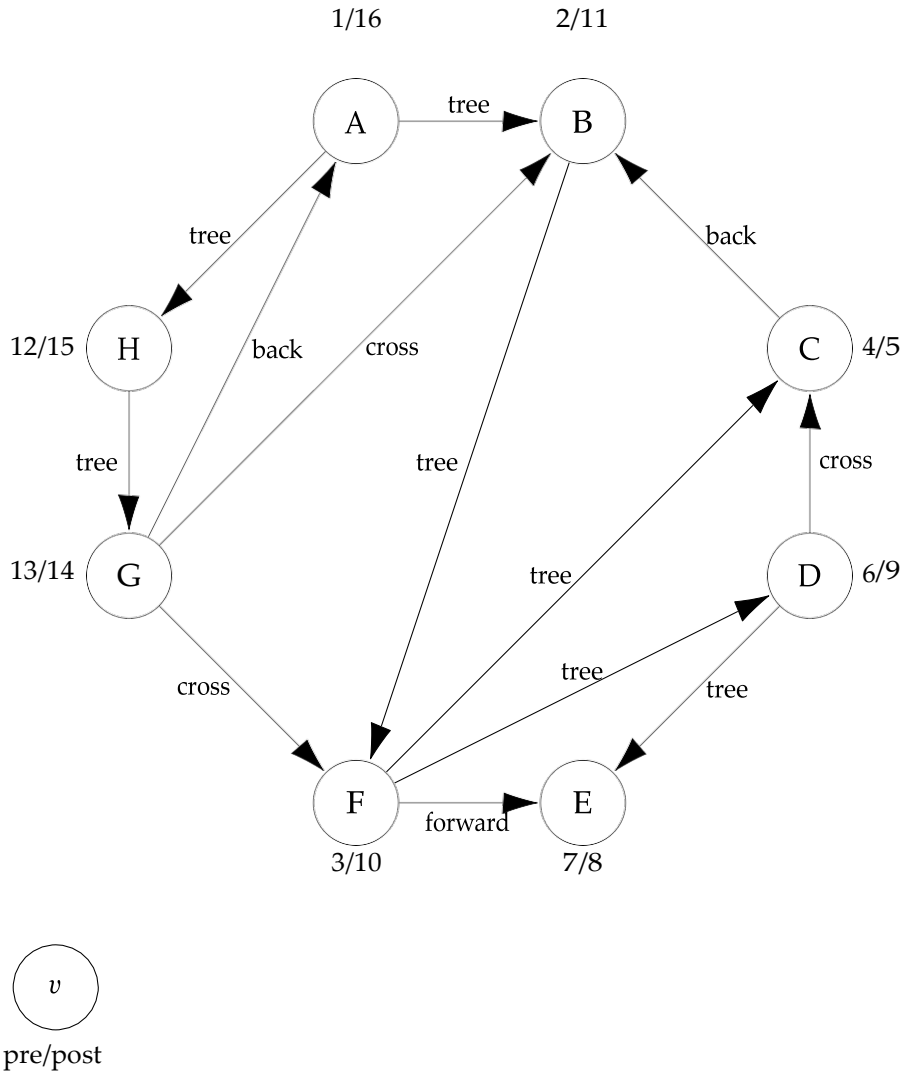
```
1:  function Explore(G, v, parent)
2:    visited(v)=true
3:    for each edge (v,next) ∈ E do
4:        if next = parent then
5:            continue
6:        if visited(v) then
7:            return true
8:        if Explore(G, next, v) then
9:            return true
10:   return false
11:
12: function FindCycle(G)
13:   for each v ∈ V do
14:       visited(v) = false
15:   for each v ∈ V do
16:       if not visited(v) then
17:           if Explore(G, v, null) then
18:               return true
19:   return false
```

Grading Criteria

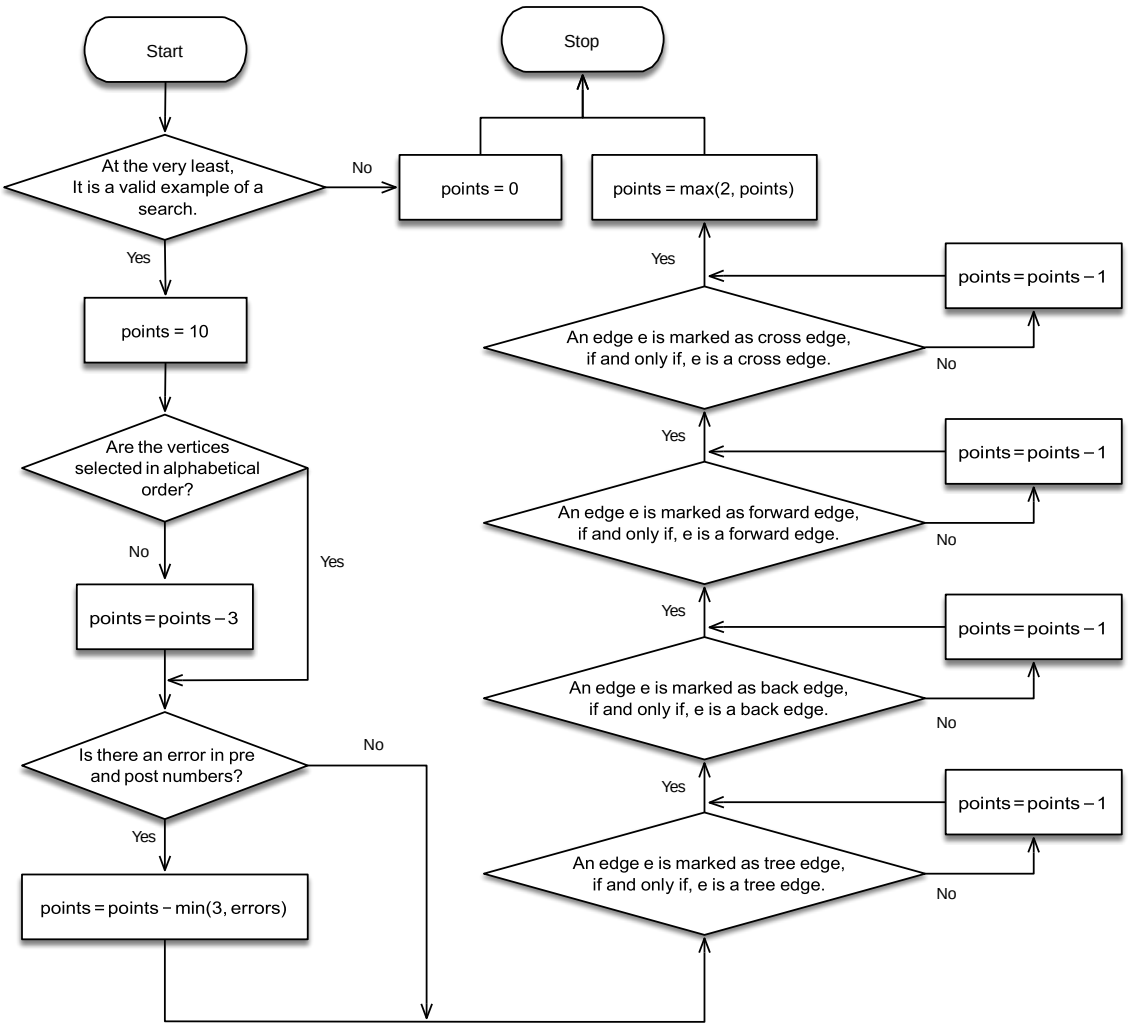
- 0 points: Mostly wrong.
- 2 points: Basic ideas are presented. However, the algorithm is incomplete.
- In other cases, see the table below.
- An additional 1-point deduction may be taken.

Time Complexity \ Correctness	Correct	Contains major errors
$O(\|V\|)$	10 points	5 points
Worse than $O(\|V\|)$	5 points	3 points

8.



Grading Criteria See the flowchart.


Start
At the very least,
It is a valid example of a
search.
No
points = 0
Stop
points = max(2, points)
Yes
points = 10
Are the vertices
selected in alphabetical
order?
No
Yes
points = points – 3
Is there an error in pre
and post numbers?
No
Yes
points = points – min(3, errors)
Yes
An edge e is marked as cross edge,
if and only if, e is a cross edge.
No
points = points – 1
Yes
An edge e is marked as forward edge,
if and only if, e is a forward edge.
No
points = points – 1
Yes
An edge e is marked as back edge,
if and only if, e is a back edge.
No
points = points – 1
Yes
An edge e is marked as tree edge,
if and only if, e is a tree edge.
No
points = points – 1