CSE317 Design & Analysis of Algorithms (Spring'19)

Final Examination

Max Marks: 50 *Time Allowed:* 3 hours

Please give <u>clear</u> and <u>rigorous</u> answers.
Be to the point. Show your work.

Name: ____________________ **ERP:** __________

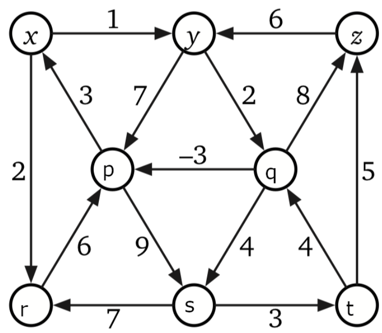
Question 1: Graphs .... *11 marks*

(a) [3 marks] Clearly indicate the following structures in the directed graph below.

i. A depth-first tree rooted at x.

ii. A breadth-first tree rooted at y.

iii. The shortest directed cycle.



(b) [4 marks] Run Bellman-Ford algorithm on the above graph starting from node z. Show `dist` values after each iteration of the main loop. Draw the computed shortest-path tree.

(c) [2 marks] Consider the execution of depth-first search on a directed graph G from vertex s, beginning with the function call $dfs(G, s)$. Suppose that $dfs(G, v)$ is called during the depth-first search. Which of the following statements can you infer at the moment when $dfs(G, v)$ is called? Mark each as True or False.

i. G contains a directed path from s to v.

ii. The function-call stack contains a directed path from s to v.

iii. If G includes an edge $v \to w$ for which w has been previously marked, then G has a directed cycle containing v.

iv. If G includes an edge $v \to w$ for which w is currently a vertex on the function-call stack, then G has a directed cycle containing v.

(d) [2 marks] Consider the execution of breadth-first search on a directed graph G, starting from vertex s. Suppose that vertex v is removed from the queue during the breadth-first search. Which of the following statements can you infer at the moment when v is removed from the queue? Mark each as True or False.

i. G contains a directed path from s to v.

ii. The queue contains a directed path from s to v.

iii. If G includes an edge $v \to w$ for which w has been previously marked, then G has a directed cycle containing v.

iv. If G includes an edge $v \to w$ for which w is currently a vertex on the queue, then G has a directed cycle containing v.

Question 2: Running Time .... *7 marks*

(a) For each of the following Java methods, choose the best matching running time from:

$$O(\log N), O(N), O(N \log N), O(N^2), O(2^N), O(N!)$$

i.

```
public static int f1(int N) {
   int x = 0;
   for (int i = 0; i < N; i++)
      x++;
   return x;
}
```

ii.

```
public static int f2(int N) {
   int x = 0;
   for (int i = 0; i < N; i++)
      for (int j = 0; j < i; j++)
         x++;
   return x;
}
```

iii.

```
public static int f3(int N) {
   if (N == 0) return 1;
   int x = 0;
   for (int i = 0; i < N; i++)
      x += f3(N-1);
   return x;
}
```

iv.

```
public static int f4(int N) {
   if (N == 0) return 0;
   return f4(N/2) + f1(N) +
      f4(N/2);
}
```

v.

```
public static int f5(int N) {
   int x = 0;
   for (int i = N; i > 0; i = i/2)
      x += f1(i);
return x;
}
```

vi.

```
public static int f6(int N) {
   if (N == 0) return 1;
   return f6(N-1) + f6(N-1);
}
```

vii.

```
public static int f7(int N) {
   if (N == 1) return 0;
   return 1 + f7(N/2);
}
```

Question 3: Reductions .... *3 marks*

Suppose that Problem X poly-time reduces to Problem Y. Mark whether each of the following statements are True or False.

(a) If X can be solved in polynomial time, then so can Y.

(b) If Y can be solved in polynomial time, then so can X.

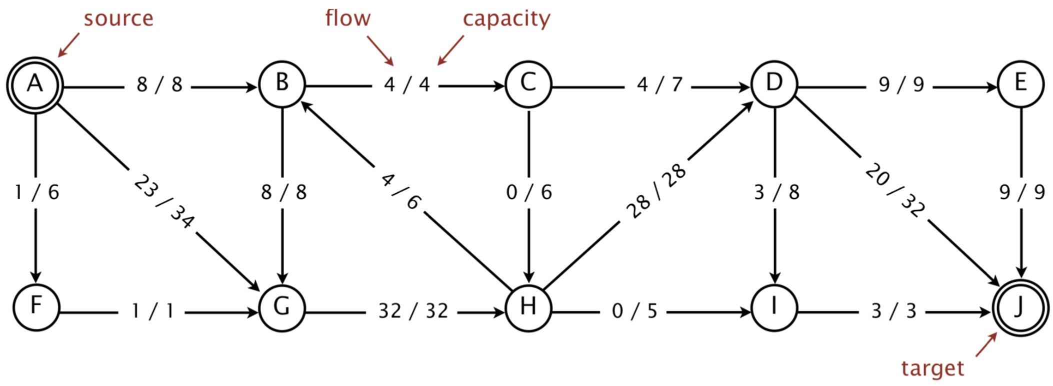
(c) If X cannot be solved in polynomial time, then neither can Y.

(d) If Y cannot be solved in polynomial time, then neither can X.

(e) If Y is NP-complete, then so is X.

Question 4: Maximum Flows .... *5 marks*

Consider the following flow network and feasible flow f from the source vertex A to the sink vertex J.



(a) [1 mark] What is the size of the flow f?

(b) [2 marks] Starting from the flow f, perform one iteration of the Ford-Fulkerson algorithm. List all vertices that are on the (unique) augmenting path. What is the bottleneck capacity of the augmenting path?

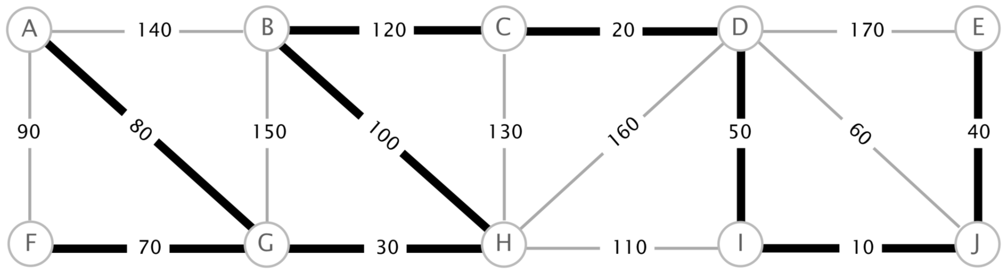
(c) [1 mark] List the vertices on the source side of a minimum cut.

(d) [1 mark] For which of the following edges, doubling the capacity would increase the value of the maximum flow?

$$A \to G, B \to C, D \to I, G \to H, I \to J, H \to D$$

Question 5: Minimum Spanning Trees .... *6 marks*

(a) Consider the following edge-weighted graph G containing 10 vertices and 17 edges. The thick black edges T define a spanning tree of G but not a minimum spanning tree of G.



i. [1 mark] Find a cut in G whose minimum weight crossing edge is not an edge in T.

ii. [2 marks] Run Prim's algorithm on G starting from vertex A to find an MST T' in G. In what order the edges of T' are found?

(b) [3 marks] Let G be any simple graph (no self-loops or parallel edges) with positive and distinct edge weights. For each of the following statements, make a short proof if the statement is true; otherwise provide a counterexample to show that it is false.

i. Any MST of G must include the edge of minimum weight.

ii. Any MST of G must exclude the edge of maximum weight.

iii. If the weights of all edges in G are increased by 17, then any MST in G is an MST in the modified edge-weighted graph.

Question 6: Divide & Conquer .... *6 marks*

(a) [3 marks] You are given a sorted array of numbers where every value except one appears exactly twice; the remaining value appears only once. Design a $O(\log n)$ algorithm for finding which value appears only once.

Here are some example inputs to the problem:

1 1 2 2 3 4 4 5 5 6 6 7 7 8 8

10 10 17 17 18 18 19 19 21 21 23

1 3 3 5 5 7 7 8 8 9 9 10 10

(b) [2 marks] State Master theorem.

(c) [1 mark] Use master theorem to solve the recurrence: $T(n) = 3T(n/2) + O(n)$

Question 7: Dynamic Programming .... *7 marks*

(a) [3 marks] In this question we will compute edit-distance between the strings X = *your-firstname* and Y = *lastname*. (If your first-name or last-name have more than 5 letters then consider the first 5 letters only).

i. Write the recurrence for computing the optimal cost of a problem given the optimal solution of relevant subproblems. How many subproblems we get?

ii. Fill in the appropriate table using the recurrence from previous part.

(b) [4 marks] Given a set of n positive integers, find if we can partition it into two subsets such that the sum of elements in both the subsets is equal. E.g., the set $\{1, 1, 3, 4, 7\}$ can be partitioned into two subsets with equal sum: $\{1, 3, 4\}$ & $\{1, 7\}$. While the set $\{2, 3, 4, 6\}$ cannot be partitioned into two subsets with equal sum.

Your algorithm shoud run in $O(nS)$ time, where S is the sum of all elements in the given input set.

Question 8: Multithreaded Algorithms .... *5 marks*

(a) [3 marks] Consider the following multithreaded pseudocode for transposing an $n \times n$ matrix A in place:

```
TRANSPOSE(A)
1: n = A.rows
2: parallel for j = 2 to n do
3:     parallel for i = 1 to j − 1 do
4:         exchange a_ij with a_ji
```

Analyze the work, span, and parallelism of this algorithm.

(b) [2 marks] Suppose that we replace the parallel for loop in line 3 of TRANSPOSE with an ordinary for loop. Analyze the work, and span of the resulting algorithm.