CSE317 Design & Analysis of Algorithms

Final Examination – Fall'17

Max Marks: 50 Duration: 3 hours

Answer all questions. Use seperate answer sheet. Be to the point. Show your work.
Please give clear and rigorous answers.

Name: _____ **ERP:** _____

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

(a) i. [2 marks] Draw a directed graph, having as its eight vertices the strings 'ape', 'ate', 'eat', 'era', 'pea', 'rap', 'rat', and 'tea', and including an edge from word x to word y whenever the last two letters of x are the same as the first two letters of y; for instance, you should include an edge from 'ape' to 'pea'.

ii. [2 marks] Write down a sequence in which the vertices of this graph could be visited by breadth first search, starting from 'era'. Also draw resulting BFS tree.

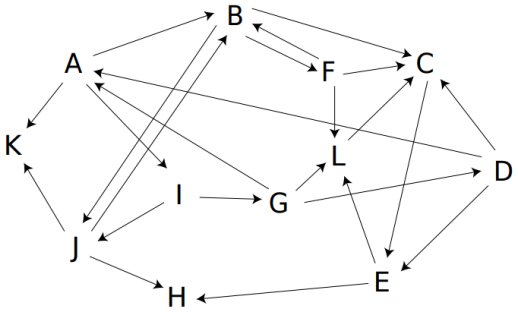
iii. [1 mark] Does this graph have a topological ordering? Explain why or why not.

iv. [2 marks] Draw an adjacency matrix representation of this graph.

(b) We say that a vertex in a directed graph is "*looping*" if it is part of a cycle of two or more vertices. For instance, in the following graph, B and J are part of a two-vertex cycle, and in fact all the vertices except for K and H are looping:

i. [2 marks] How is the property of being looping related to strongly connected components?

ii. [2 marks] Describe a linear-time algorithm that lists all of the looping vertices of a graph.



Question 2: Shortest paths ..... *7 marks*

(a) [4 marks] State the worst-case running times of following shortest-path algorithms, on graphs with A vertices and B edges. Your answer should be expressed using O-notation, as a function of A and B.

i. The Bellman-Ford algorithm

ii. Dijkstra's algorithm

iii. The topological-sorting DAG shortest path algorithm

For graphs on which all three algorithms can be used, which one has the fastest time?

(b) [3 marks] In ancient ages, there used to be no currency. People used to trade using barter. For example, an apple may be exchanged with 1.5 sack of rice; one sack of rice may be exchanged with 2.5 bags of potatoes etc. You have a directed graph $G = (V, E)$ where the nodes represent the items used in barter (1 apple, 1 sack of rice, 1 bag of potatoes etc.) and an edge between the two nodes represents the exchange rate between the two items (like, the weight of the edge from 1 apple to 1 sack of rice would be 1.5). You can assume that the barter rates are consistent between any two items (like, if the edge from 1 sack of rice to 1 bag of potato has the weight 2.5, the reverse edge would have a weight of 0.4).

You are a businessman and you want to make profit just by exchanging goods. Find if it is possible for you to start with some X amount of one item, keep on exchanging it into various other items and finally end up with $> X$ amount of the original item.

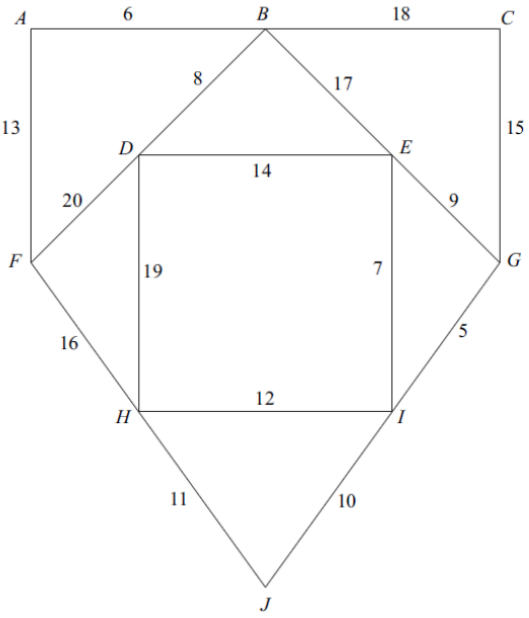
Question 3: Greedy Algorithms ..... *7 marks*

(a)

i. [1 mark] State the number of edges in a minimum spanning tree (MST) of a graph with n vertices.

ii. [2 marks] Run Kruskal's algorithm on the graph shown on right. Write the edges in the order which they are added to the minimum spanning tree and draw the final MST.

iii. [1 mark] What is the total weight of minimum spanning tree in this graph?



(b) [3 marks] Suppose you have an undirected graph with weighted edges, and perform a depth-first search, such that the edges going out of each vertex are always explored in order by weight, smallest first. Is the depth first search tree resulting from this process guaranteed to be a minimum spanning tree? Explain why, if it is, or, if it isn't, provide a counterexample (specifying the start vertex for the DFS and showing both trees).

Question 4: Divide & Conquer ..... *9 marks*

(a) [3 marks] You are the TA for a class with an enrollment of n students. You have their final scores (unsorted), and you must assign them one of the G available grades (A, B, C etc.). The constraints are (assuming n is a multiple of G):

- Exactly (n/G) students get each grade (for example, if $n = 30$, and $G = \{A, B, C\}$, then exactly 10 students get A, 10 get B, and 10 get C)
- A student with lower score doesn't get a higher grade than a student with a higher score (however, they may get the same grade)

Assuming that each student received a different score, derive an efficient algorithm and give its complexity in terms of n and G. Any algorithm that first sorts the scores will receive zero credit.

(b) The following recursive algorithm sorts a sequence of n numbers.

```
def triplesort(seq):
    if n <= 1: return
    if n == 2:
        replace seq by [min(seq),max(seq)]
        return
    triplesort(first 2n/3 positions in seq)
    triplesort(last 2n/3 positions in seq)
    triplesort(first 2n/3 positions in seq)
```

i. [3 marks] Write down a recurrence describing the running time of the algorithm as a function of n. Solve your recurrence using Master theorem.

ii. [3 marks] How would you parallelize the above program in the multi-threading framework discussed in the class? Give Work (T_1), Span (T_∞), and Parallelism of the resulting program.

Question 5: Dynamic Programming ..... *7 marks*

(a) [3 marks] Suppose your job is to sell advertising space on a web page. At most half the screen can be filled with ads, and for each potential ad buyer we can associate a pair of numbers (f, p) where f is the fraction of a screen the buyer's ad would take and p is the price he's willing to pay for his ad. Your job is to choose a subset of the ads that maximizes your total profit. Ads can not be resized; you must take them at the given size or not take them at all. Would this problem be more naturally modeled as a fractional knapsack problem, or a 0-1 knapsack problem? Explain your answer.

(b) [4 marks] You are given a sorted array of n distinct integers of sum S (S is even). Design a Dynamic programming algorithm to decide if the elements of the array could be partitioned into two groups of equal sum.

For example, $[1, 3, 4, 6, 8, 14]$ can be partitioned into $(1, 3, 6, 8)$ and $(4, 14)$ while $[3, 4, 11, 16]$ cannot be partitioned.

Question 6: Big-O notation & NP-Completeness ..... *9 marks*

(a) [4 marks] For each pair of expressions $T(n)$ and $f(n)$ below, indicate whether $T(n)$ is O, Ω, or Θ of $f(n)$.

$T(n)$	$f(n)$	$T(n) \stackrel{?}{=} O(f(n))$	$T(n) \stackrel{?}{=} \Omega(f(n))$	$T(n) \stackrel{?}{=} \Theta(f(n))$
$\log 3^n$	$\log 2^n$			
$10n \log 10n$	$n \log n$			
$\sqrt{n}$	$(\log n)^3$			
2^n	2^{n+1}			

(b) Answer in short (justification must be given for each problem, a yes or no answer is not sufficient):

i. [2 marks] Manna and Jimin both submit their PhD theses.

- Jimin claims to have found out a polynomial time algorithm to decide SAT.
- Manna claims to be have found out a $o(n \log n)$ time comparison-based sorting algorithm to sort n arbitrary numbers.

Who is more likely to get his/her PhD?

ii. [2 marks] Suppose problem A can be reduced into problem B in $\Theta(n^3)$ time. Problem B is known to have an $O(n^4)$ algorithm. Choose the strictest bound of running time of A using B from the following three options:

- $\Theta(n^3) + O(n^4) = O(n^4)$
- $\Theta(n^3) * O(n^4) = O(n^{3+4}) = O(n^7)$
- $\Theta(n^3) \circ O(n^4) = O(\Theta(n^3)^4) = O(n^{12})$

iii. [1 mark] True or False? It is known that there can be no polynomial-time algorithm for a NP-complete problem.