Algorithms
Comprehensive Exam
(Spring 2019)

SHORT QUESTIONS **(Answer all six questions, each carrying 7 points.)**

1. Give a big-O (upper bound) estimate for $f(n) = nlog(n!) + 3n^2 + 2n - 10000$, where n is a positive integer.

2. The hash table is a widely adopted data structure. Explain briefly how perfect hashing works. Separately, what is the situation when a new key cannot be inserted in a Cuckoo hash table successfully?

3. Show your construction of an optimal Huffman code for the set of 10 frequencies: **a**:2 **b**:6 **c**:5 **d**:8 **e**:13 **f**:21 **g**:34 **h**:15 **i**:27 **j**: 9. 36

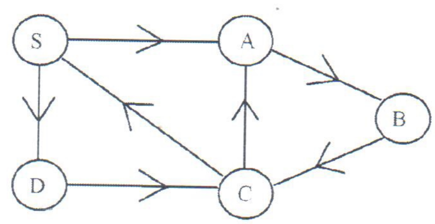
70

a c b d j e h f i g

34 2 5 6 8 9 13 15 21 27 34

4. Given a weighted directed graph $G = (V, E, w)$ and a shortest path P from s to t, if the weight of every edge is doubled to produce $G^* = (V, E, w^*)$, is P still a shortest path in G^*? Explain your reasoning behind your answer.

5. BFS (breadth-first search) and DFS (depth-first search): Give the visited node order for each type of graph search, starting with S below.

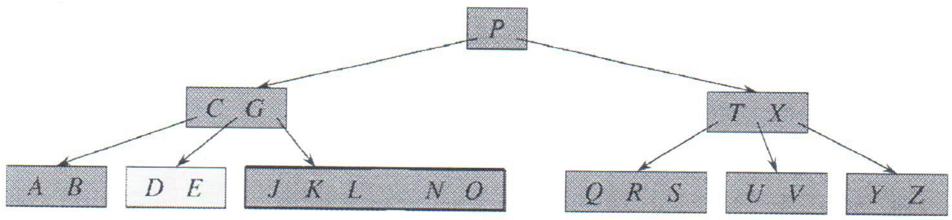


6. Many problems have been proved to be NP-complete. To prove NP-completeness, it is necessary in general to demonstrate two proof components. What are the two proof components to show a problem being NP-complete?

   Being NP-complete, the traveling-salesman problem (TSP) has a 2-approximation solution in polynomial time based on establishing a minimum spanning tree (MST) rooted at the start/end vertex (in polynomial time following MST-PRIM), if the graph edge weights observe triangle inequality. Sketch a brief proof to demonstrate that such a solution satisfies 2-approximation.

LONG QUESTIONS (Answer all four questions, each carrying 15 points.)

1. Given a B-tree with the minimum degree of $t = 3$ below, show the results after (i) deleting B, (ii) followed by inserting M, (iii) then followed by deleting T, and then (iv) inserting M_1, for $M < M_1 < N$.



2. Knapsack problem: suppose you want to pack a knapsack with weight limit W. Item i has an integer weight w_i and real value v_i. Your goal is to choose a subset of items with a maximum total value subject to the total weight $\leq W$.

   Let $M[n, W]$ denote the maximum value that a set of items $\in \{1, 2, \ldots, n\}$ can have such that the total weight is no more than W. We have the following recursive formula:

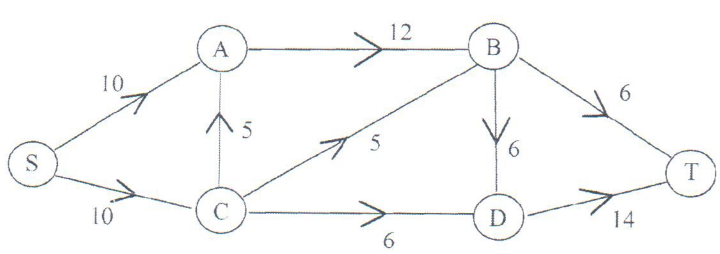
$$M[n, W] = \begin{cases} 0 & if\ n = 0\ or\ W = 0 \\ M[n-1, W] & if\ w_n > W \\ \max\{M[n-1, W - w_n] + v_n, M[n-1, W]\} & otherwise \end{cases}$$

   The time complexity of a simple recursive procedure as given below is exponential.

```
M(n, W)
  {
     if (n == 0 or W == 0) return 0;
     if (w_n > W)
           result = M(n -1, W)
     else
           result = max{ v_n + M(n -1, W - w_n), M(n -1, W)};
     return result;
  }
```

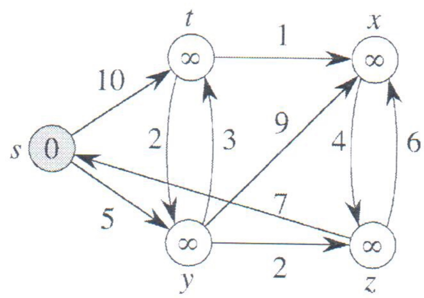
   Provide a dynamic programing solution of Knapsack problem after adding two lines of code to the above procedure. (Hint: use a table to memorize the results.)

3. Follow Ford-Fulkerson Algorithm to compute the max flow of the flow network illustrated below. Show each step to compute the max flow and also show the min cut of the flow network.



4. The Dijkstra's algorithm (*DIJ*) solves the single-source shortest-path problem in a weighted directed graph $G = (V, E)$. Given the graph G below, follow *DIJ* to find shortest paths from vertex s to all other vertexes, with all predecessor edges shaded and estimated distance values from s to all vertexes provided at the end of each iteration.

What is the time complexity of *DIJ* for a general graph $G = (V, E)$, if candidate vertexes are kept in a binary min-heap?



$O((V+E) \lg V)$

Good Luck!