Algorithms
Comprehensive Exam
(Spring 2022)

SHORT QUESTIONS **(Answer any five questions, each carrying 8 points.)**

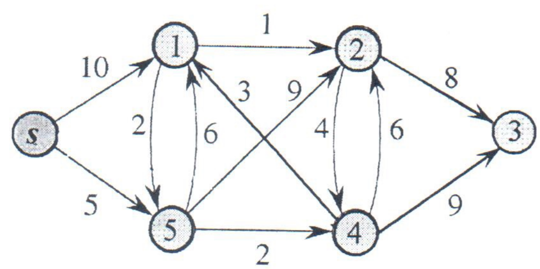
1. Solve the two recurrence expressions below:

$$T_1(n) = 2\cdot T(n/4) + T(n/2) + n^{1/3}$$

$$T_2(n) = T_2(n/3) + T_2(2n/3) + n\cdot \lg(n)$$

2. Follow depth-first search (*DFS*), starting from Node *s*, to traverse all nodes of the graph shown below. Mark (a) the type of every edge and (b) the discovery and the finish times of each node.

   How do we utilize the *DFS* result for quickening the solution of single-source shortest paths in a direct acyclic graph?



3. Given that for an open-address hash table with load factor $\alpha = n/m < 1$, the expected number of probes in unsuccessful search under uniform hashing is at most $1/(1-\alpha)$, prove the expected number of probes in a successful probe under uniform hashing being at most $(1/\alpha)\cdot \ln(1-\alpha)^{-1}$ by giving a proof sketch which explains how many probes are needed to locate existing keys.

4. A binary search tree (T) is to be maintained following the in-order tree traversal order. Consider a sequence of arrival keys, {25, 23, 14, 7, 9, 21, 31, 34, 18, 24, 19, 5}, to T which has just the root node with its key = 20 initially.

   (a) Show the resulting T after inserting all arrival keys.

   (b) Show the resulting T after its root node is then deleted.

5. How many ones does the following code print when running with input n? Compute the exact value, if possible; otherwise, provide a big-O bound.

(a) Ones(n):

```
if n ≤ 0
        { print 1 }
else
        { Ones(n-1)
          Ones(n-2) }
```

(b) Ones(n):

```
if n = 0
        { print 1 }
else
        for i = 1 to n
                { Ones(n-1) }
```

(c) Ones(n):

```
if n = 0
        { print 1 }
else
        for i = 1 to 2^n
                { Ones(n-1) }
```

6. Circle (T)rue or (F)alse for statements below, without justifying your choice.

a. T F Let T be a complete binary tree with n nodes. Finding a path from the root of T to a given vertex v using breadth-first search takes O(lg n) time.

b. T F Given a weighted directed graph G = (V, E, w) and a shortest path P from Node s to Node t, if we added the same weight to every edge to produce G* = (V, E, w*), then P is still a shortest path from s to t in G*.

c. T F A directed acyclic graph may have multiple different topological orderings.

d. T F Given a graph G = (V, E) with positive edge weights, the Bellman-Ford algorithm and Dijkstra's algorithm can produce different shortest paths despite always producing the same shortest-path weights.

e. T F Knapsack problem is not an NP-Complete Problem because it can be efficiently solved using dynamic programming technique.

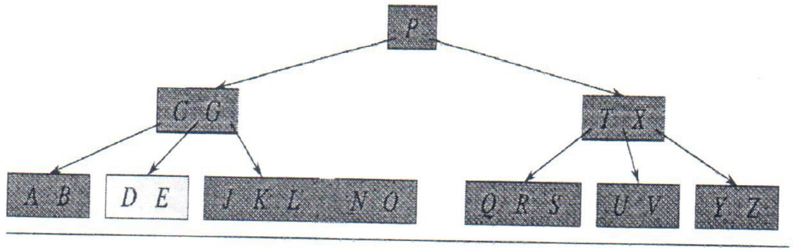
f. T F Given a connected graph G = (V, E), if a vertex v ∈ V is visited during level k of a breadth-first search from source vertex s ∈ V, then every path from s to v has length at least k.

g. T F If a dynamic-programming problem satisfies the optimal-substructure property, then a locally optimal solution is globally optimal.

h. T F Every problem in NP can be solved within exponential time.

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

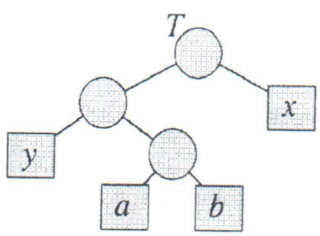
1. Given a B-tree with the minimum degree of $t = 3$ below, show the results after (a) deleting V, (b) then followed by inserting M, (c) then followed by deleting B, and (d) then followed by deleting S.



2. The Floyd-Warshall algorithm (FW) obtains all pairs of shortest paths in a weighted directed graph. Consider the graph given in Problem SHORT-2 above, with Vertice s ignored. What is the recursive equation of $d_{i,j}^{(k)}$ for the shortest-path weight of any path between i and j with intermediate vertices $\in \{1, 2, 3, \ldots, k\}$?

   Derive all distance matrices $D^{(k)}$ following FW so that the $d_{i,j}^{(n)}$ element of final matrix $D^{(n)}$ denotes $\delta(i, j)$ for every vertex pair $<i, j>$ for all $i, j \in \{1, 2, 3, 4, 5\}$.

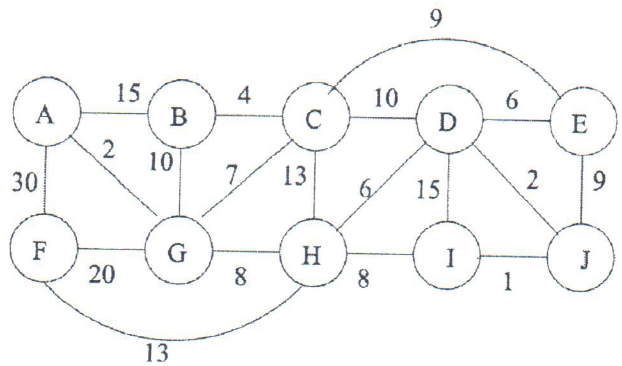
3. Sketch a proof of the Lemma below, using the tree provided.
   Let C be an alphabet in which each character $c \in C$ has frequency $c.freq$. Let x and y be two characters in C having the lowest frequencies. Then there exists an optimal prefix code for C in which the codewords for x and y have the same length and differ only in the last bit.



4. A rabbit is running up a staircase with 10 steps and can hop either 1 step or 2 steps at a time. How many possible ways the rabbit can hop up the stairs? Please give the exact number.

   If a staircase has n steps, please show your algorithm to compute all the possible ways the rabbit can hop up the stairs.
   (Hint: apply dynamic programming.)

5. Given the un-directed graph below, run Dijkstra's algorithm, starting at Vertex A. Note that the algorithm works in the same way on an un-directed graph as on a directed graph.



Show each step by filling out the table. The second column denotes the set S, which refers to the nodes whose shortest distances from A have been determined. The third to twelfth columns show the shortest distances from A to other vertices. Add rows when needed.

	Set S	A	B	C	D	E	F	G	H	I	J
initialization											
1st iteration											
2nd iteration											

Good Luck!