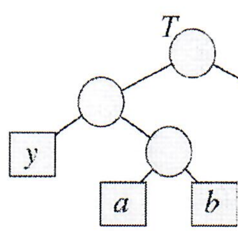
Algorithms
Comprehensive Exam
(Spring 2018)

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

1. A problem with size n follows a typical divide-and-conquer approach to have its time complexity of $T(n) = T(n/4) + T(3n/4) + c{\cdot}n$. Solve $T(n)$. (Show your work)

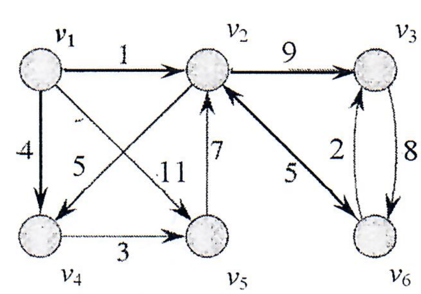
2. Given that for an open-address hash table with load factor $\alpha = n/m < 1$, the expected number of probes in unsuccessful search under uniform hashing is at most $1/(1 - \alpha)$, how do you prove the expected number of probes in a successful probe under uniform hashing being at most $(1/\alpha)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

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. The Dijkstra's algorithm (DS) solves the single-source shortest-path problem in a weighted directed graph $G = (V, E)$ without negative weighted edges or cycles, by edge relaxation at one vertex at a time until all vertices are examined. Given the graph G below, follow DS to find shortest paths from vertex v_1 to all other vertices, with all predecessor edges shaded and estimated distance values from v_1 to all vertexes provided at the end. Also list the sequence of vertices at which relaxation takes place.

   What is the time complexity of DS for a general graph $G = (V, E)$, when candidate vertices are kept in an array?



5. (a) Define height balanced binary tree.

   (b) Write a pseudo code to determine whether a tree is height balanced?

   (c) Obtain the tight bound of your algorithm.

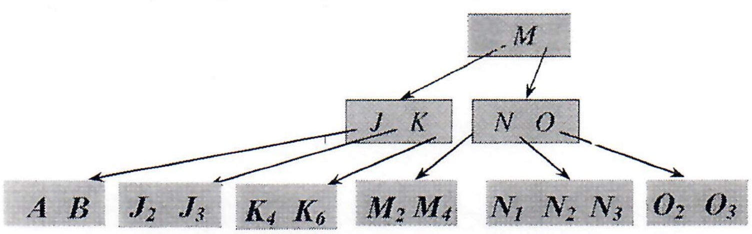
6. (a) What are the properties of min heap and max heaps?

   (b) What is the preferred data structure of implementing a binary heap? (Justify your answer.)

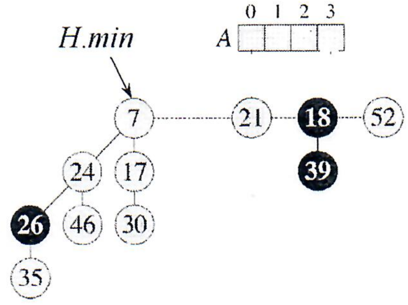
   (c) What is the time complexity of merging two different min heaps sized with n and m.

<u>LONG QUESTIONS</u> (Answer any three questions, each carrying 20 points.)

1. Given the initial <u>B-tree</u> with the minimum node degree of <u>$t = 3$</u> below, show the results (a) <u>after deleting</u> the key of M_2 , (b) followed by inserting the key of L, (c) then by deleting the key of J_2, (d) then by <u>inserting</u> the key of O_1, with $O < O_1 < O_2$, and (e) then by deleting K. (Show the result after each deletion and after each insertion.)



2. A <u>Fibonacci min-heap</u> relies on the procedure of CONSOLIDATE to <u>merge min-heaps</u> in the root list upon the operation of extracting the minimum node. Given the following Fibonacci min-heap, show <u>every consolidation step</u> and the <u>final heap result</u> after $H.min$ is <u>extracted</u>, with the aid of A.



3. (a) To what extent the asymptotic upper bound and lower bound provide insight on running time of an algorithm.

   (b) Compare and contrast asymptotic tight bound to average running time of an algorithm.

   (c) Consider the pseudo code of an algorithm given below.
      c.1 What does the value K in Line 4 denote?
      c.2 What does the value m in Line 8 denote?
      c.3 When the algorithm terminates, what does the value of m+K in Line 9 denote?
      c.4 Find the asymptotic tight bound of Algorithm Test below.

```
  Algorithm test (n)
  K ← 0
2 for i = 1 to n
3     for j = 1 to i
4         K = K + 1
5 m = 0
6 for i = 1 to n-1
7     for j = i+1 to n
8         m = m + 1
9 return (m+K)
```

4. (a) Define the following classes of a decision problem: P, NP, and NP-completeness.

   (b) Consider the 0-1 knapsack problem with n objects each with its respective pre-defined profit. The objective is to maximize the total profit that can be accommodated into a container of capacity W. Defining appropriate notations for weight and profit of objects, formulate the problem.

   (c) Convert of the problem that you have defined in (b), into a decision problem.

   (d) Show the problem that you have defined in (c) belongs to NP-class.

   (e) Does the problem in (d) belong to the P-class or NP-completeness. (Justify your answer.)

   (f) If principle of optimality be applicable to solve the problem defined in (c), formulate it. Otherwise, explain why not.

   (g) What would be your explanation, if 0-1 knapsack problem is solved by dynamic programming in polynomial time?