Algorithm and Theory of Computation

Short Questions

Answer all the questions

1. What does it understood by time and space complexity? Do these concepts still remain to be relevant in the era of massive parallelism and scalable computing such as hadoop? (if your answer is yes explain why otherwise why not)

2. The time complexity of algorithm 1 and algorithm 2 to solve the same problem is given respectively $O(n^2)$ and $O(n^3)$.

   a. Which algorithm would you choose to solve the problem.
   b. If the running time of algorithm1, and algorithm 2 are respectively $T(n) = 1000\ n^2$ and $T(n) = 2\ n^3$, which algorithm would you choose. Explain your rationale.

3. Briefly describe when divide and conquer strategy can be applied to solve problem?

B. Can you effectively apply divide and conquer strategy to solve the following problems (explain why or why not)
I. minimum spanning tree
II. 0-1 knapsack problem
III. Merge sort

4. Find the tight bounds of the following sums.
$\sum_{i=1}^{i=n} i^k$ where k is a constant

5. a. Briefly describe what is understood by union-find problem?
   b. How does binomial tree helps to solve union find problem.
   c. How many nodes are there in a binomial tree of order k.
   d. If you remove a node from a binomial tree of order k, how many binomial trees will be created?

6. Briefly describe priority queue. What is the preferred data structure for implementing priority queue? Obtain the time complexity of building a propriety queue of N integers.

7. Briefly describe what is dynamic programming. Can you apply this technique to solve any optimization problem? If yes explain why, if not what particular conditions must be met.

8. What is the strategy behind greedy algorithm? Will it always provide an optimal solution? Explain why and why not.

b. What are the differences between Prim's and Kruskal's algorithm in solving minimum spanning tree?

9. Mark true/false (T/F) against the following statements:
   1. A binary search tree of size N will always find a key at most O(log N) time.
   2. A breadth first search can be considered as a special case of heuristic search algorithm
   3. An optimal binary search tree is not necessarily being a balanced tree.
   4. The dynamic programming approach uses top-down strategy to solve optimization problem.

Long Questions

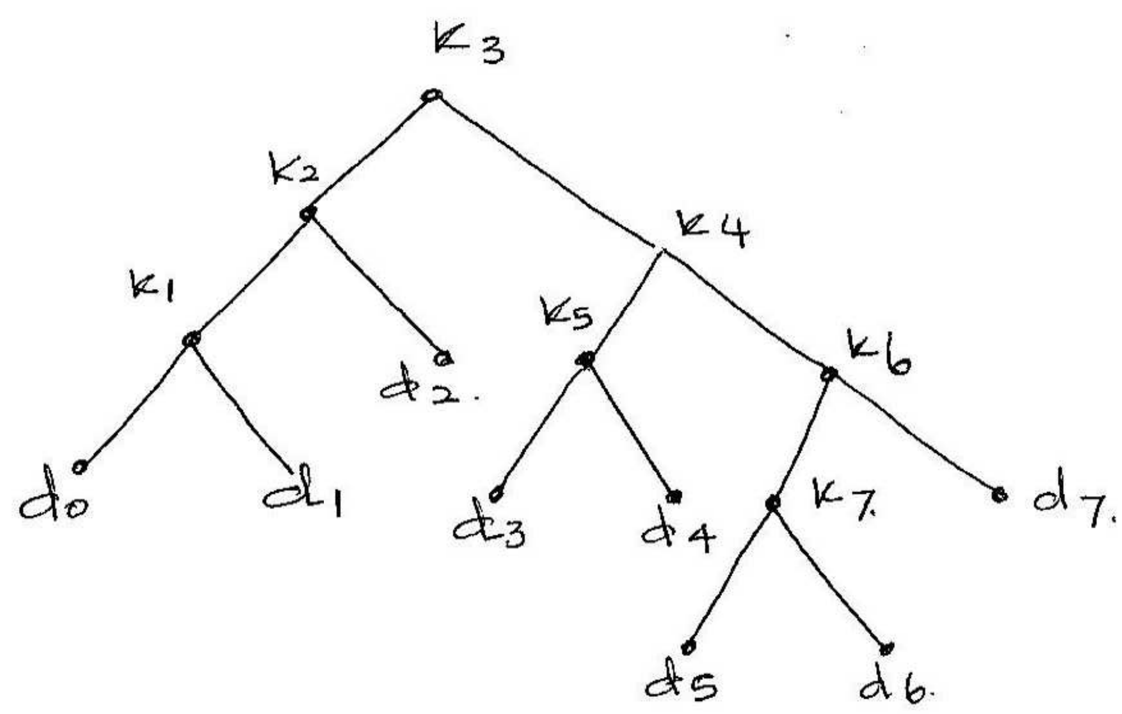
Answer any 3 of 4 questions.

1. a. An object r is accessed by its key k_r. If all the objects have an equal chanced of being accessed, what data structure will help you to have a better tight bound? (State your assumptions and provide the bound you have obtained for the proposed structure)

b. Describe how dynamic program is used to construct an optimal binary search tree.

b. Using the following probability of p_i and q_i, obtain the expected cost of searching the binary search tree of Figure 1. q_i denotes the probability of accessing d_i

i	0	1	2	3	4	5	6	7
p_i		0.04	0.06	0.08	0.06	0.1	0.12	0.1
q_i	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05



2.A. Explain what do you understand by "principle of optimality"

B. Write down the basic rules that satisfy the principle of optimality and domain related constraints to the following problems.

B1. 0-1 knapsack problem.
B2. Pairwise shortest path problem.
B3. Chain matrix multiplication problem.

3. Compare and contrast P, NP, NP-complete and NP-hard
b. Based on current conjecture, draw a venn diagram to show the relationship among these classes of problem.

c. Suppose there are n clauses and m propositions in a given 3p-sat problem. How many possible interpretations are there? What is the time complexity of testing the satisfiability of a given interpretation? What is the time and space complexity of testing the satisfiability of the clauses.

d. Suppose a single NP-complete problem is solved in polynomial algorithm, what can you state about the entire NP-complete class as well as the NP-hard class.

4. Classify each of the following language as regular, context free but not regular, or decidable but not context free. Prove your answers.

a. $\{ a^{n+1} b^{n-1} c^{m} : n,m > 0\}$
b. $\{ a^{2n} b^{2m+1} : n, m >= 0\}$
c. $\{ a^{n+1} b^{n-1} c^{n} : n, m >= 0\}$