Sample Data-Structures and Algorithms Questions for the Computer-Science Fundamental Areas Exam

Disclaimer: The following questions are not in any way necessarily indicative of the phrasing or difficulty of the questions on the comps. These are simply questions provided to aid your studying as a reminder of the subject matter. By posting this, we are NOT implying that this list of questions is at all comprehensive."

1. Insert the following numbers into a binary search tree in the order that they are given and draw the resulting tree.

   $$87, 36, 22, 15, 56, 85, 48, 90, 72, 6$$

   Delete 48 and draw the resulting tree. Delete 15 and draw the resulting tree. Repeat the above exercise for AVL trees, Red-black trees, and splay trees.

2. A full binary tree is one in which every internal (i.e. non-leaf) node has 2 children. How many internal nodes are there in a full binary tree with N total nodes?

3. What is the worst-case asymptotic time complexity of an insert operation into an N node splay tree?

4. Show the steps of quicksort on the following list of unsorted integers. Assume that the pivot node is always the one on the lefthand side of the list. Box all pivots and underline all numbers that have been placed in their correct respective positions.

   $$87, 36, 22, 15, 56, 85, 48, 90, 72, 6$$

5. Show that mergesort has time complexity $O(n \log n)$.

6. Radix sort the following list of 5 digit numbers showing each step in the sort.

   $$17725, 82358, 12347, 45902, 93, 83976, 23789$$

   What is the asymptotic running time for a radix sort on n numbers with d digits each?

7. Add the following list of numbers to an initially empty binary heap: 12, 5, 15, 9,13, 7, 15, 10, 3, 20, 4.

8. For the heap in the previous problem, show the resulting heap after deleting the first five least elements.

9. What is the worst-case complexity of inserting n elements into an initially empty heap, assuming the elements are inserted one at a time. Compare this with the complexity of buildHeap().

10. If the load factor of a hash table is currently .8, then using random probing, how many collisions should we expect on the average before an open address is found?

11. All of the data entered into the following hash table are required to have final digit between 0 and 6. Suppose keys 1121, 1432, 1321, 1323, 1841, 1223, are hashed into a table of size 7, using the hash function $h(x) = x$ mod 10, and linear probe function $f(i) = ai + b$, for nonnegative integer constants a and b. Solve for a and b assuming the resulting table below (keys were hashed in the order of which they appear above). Assume that $h(k) + f(0)$ is used to probe for the first open address in the event of a collision.

cell 0	**cell 1**	**cell 2**	**cell 3**	**cell 4**	**cell 5**	**cell 6**
empty	1121	1432	1323	1223	1841	1321

12. Suppose that a hash table has load factor λ. Then if separate chaining is used to resolve collisions, what is the average length of a chain if the hash function ideally distributes keys uniformly throughout the table. Hint: chains of length zero are allowed.

13. Let D be a container-class data structure that holds nodes of a graph. Consider the following algorithm for traversing nodes of the graph. Step 0: begin with a start node s. Mark it as "visited" and place it in D. Step n ($n > 0$): remove the "next" (where "next" depends on the order in which data exits D) node n from D. For each neighbor $\hat{n}$ of n that has yet to be visited, mark $\hat{n}$ as "visited" and place $\hat{n}$ in D. What type of graph traversal does one get if D is a queue? stack?

14. Does the function $\frac{1}{1-\frac{1}{n^{\frac{1}{\sqrt{n}}}}}$ have polynomial growth?

15. Asymptotically speaking what is the relationship between $f(n) = n!$ and $g(n) = 2^{n^2}$. For example, does $g(n) = \Theta(f(n))$?

16. Give an example of a function that has superpolynomial growth, but not exponential growth. Verify that your example is correct.

17. Give an example of a functions $f(n)$ and $g(n)$ for which both $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$.

18. What is the asymptotic complexity of $\sum_{i=1}^{n} \log i$?

19. What is the asymptotic complexity (in terms of n and k for the code shown below.

```
int unknownAlgorithm(int a[], int k, int n){
    int i,j, mini,tmp;

    for(i=0; i< k; i++){
        mini = i;
        for(j= i+1; j < n; j++)
            if(a[j] < a[mini])
                mini=j;
        tmp = a[i];
        a[i]=a[mini];
        a[mini]=tmp;
    }
    return a[k-1];
}
```

20. You are developing a divide-and-conquer algorithm which must have asymptotic complexity $O(n \log n)$ to be of practical use. Moreover, you have decided to divide the problem into 3 subproblems of size $n/4$ each, where n is the size of the original problem. Using this strategy is it possible to achieve the desired complexity? If so, what is the most number of steps (as a function of n) that may be used for dividing up the original problem and combining the solutions of the subproblems? Explain.

21. Given the recurrence relation $S(n) = 2S(n/2) + n \log n$, where $S(n)$ denotes the number of steps an algorithm requires for an input of size n. What is the asymptotic complexity of the algorithm?

22. Given the recurrence relation $S(n) = 9S(n/3) + 5n^2 + 3n + 6$, where $S(n)$ denotes the number of steps an algorithm requires for an input of size n. What is the asymptotic complexity of the algorithm?

23. Give the asymptotic growth rate of $S(n)$, the number of steps required by the following code.

```
sum=0;
for(i=0; i < n; i++){
    j=i;
    while(j !=0){
        if( j % 2 == 0)
            sum++;
        j /= 2;
    }
}
```

24. Write a recursive algorithm for finding the height of a binary tree. The input to the algorithm is the node of a tree. Moreover, you may assume that each node has pointers to its left and right children.

25. Write a recursive version of the binary search algorithm.

26. Write an algorithm which inputs a simple graph and outputs whether or not the graph is connected.

27. Draw the *weighted directed* graph $G = (V, E, c)$, where the edges/costs are given by

$$E = \{(a, b, 10), (a, d, 5), (b, c, 1), (b, d, 2), (c, e, 4), (d, b, 3), (d, e, 2), (d, c, 9), (e, c, 6)\}.$$

For this graph make a table showing each stage of Dijkstra's algorithm for finding minimum cost paths from vertex a to every other vertex in the graph.

28. Draw the *directed* network $G = (V, E, c, s, t)$, where the edges/capacities are given by

$$E = \{(s, b, 10), (s, d, 5), (b, c, 5), (b, d, 2), (c, e, 4), (d, b, 3), (d, e, 2), (d, c, 9), (e, c, 6), (c, t, 10), (e, t, 7)\},$$

Where s is the source and t is the sink. What is the maximum amount of flow that can leave s and reach t.

29. Given a tree $T = (V, E, r)$ the internal path length (IPL) of T is defined as

$$IPL = \sum_{v \in V} depth(v).$$

Write a recursive algorithm for finding the IPL of a binary tree.

30. Draw the simple weighted graph $G = (V, E, w)$, where the edges/weights are given by

$$E = \{(a, b, 1), (a, c, 3), (b, c, 3), (c, d, 6), (b, e, 4), (c, e, 5), (d, f, 4), (d, g, 4),$$
$$(e, g, 5), (f, g, 2), (f, h, 1), (g, h, 2)\}.$$

Use Prim's algorithm to find a miniumum spanning tree for the graph. Do the same using Kruskal's algorithm.

31. A third algorithm for finding minimum spanning trees has one successively remove the edge of *greatest* weight, so long as the removal of the edge does not disconnect the graph. Why might such an algorithm not be as preferable to Prim's and Kruskal's?

32. Given the five matrices below, using dynamic programming find a full parenthesization of $A_1, \ldots, A_5$ with minimum multiplication complexity.

matrix	dimension
A_1	3×2
A_2	2×5
A_3	5×2
A_4	2×4
A_5	4×1

33. Suppose you have at your disposal a Turing machine M which can decide in polynomial time whether or not a graph $G = (V, E)$ has a clique of size k, for some $0 < k \leq |V|$. Describe in one or more paragraphs how to use M for finding an actual k-clique for G in polynomial time, assuming that one exists.Assuming the computational complexity of M is $p(n)$, what is the complexity of your algorithm?

34. Write a (graphical) program for a Turing machine that accepts all binary strings with even parity. Note: you may assume that the empty string has even parity.

35. An instance of Subset Sum is a (multi) subset $S \subset N$ of natural numbers, and a natural number t. The problem is to decide if there are elements $s_1, \ldots, s_k \in S$ for which $s_1 + \cdots + s_k = t$. An instance of Set Partition is a (multi) subset $S \subset N$ of natural numbers. The problem is to decide if S can be partitioned into two disjoint sets A and B such that

$$\sum_{s \in A} s = \sum_{t \in B} t.$$

Use Subset Sum to prove that Set Partition is NP-complete.

36. Use 3-SAT to prove that k-CLIQUE is NP-complete.

37. Consider the Traveling Salesperson Problem (TSP) in which one is asked to find a tour of n cities in such a way such as to minimize the total travel cost (we assume a matrix M, where m_{ij} denotes the cost of traveling from city i to city j). In one or more paragraphs, describe a greedy algorithm for finding an approximate solution to TSP. Give an example that demonstrates that your algorithm does not always find the optimal tour.

38. The divide-and-conquer recurrence relation for algorithm A is given as

$$S(n) = S(n/4) + f(n),$$

where $f(n) = \Theta(1)$. Use the Master Theorem to determine the asymptotic complexity of A.

39. The sides of a pentagon and its diagonals have costs given by the following matrix having rows and columns 0-4.

$$\begin{pmatrix} \_ & 1 & 5 & 6 & 7 \\ \_ & \_ & 1 & 5 & 2 \\ \_ & \_ & \_ & 2 & 5 \\ \_ & \_ & \_ & \_ & 11 \\ \_ & \_ & \_ & \_ & \_ \end{pmatrix} \tag{1}$$

For example, entry 0,3 means that the diagonal connecting vertex 0 to vertex 3 has a cost of 6. Assume that the sides of the pentagon are $(0,1),(1,2),(2,3),(3,4),(0,4)$. Let $t(i,j)$ denote the optimal cost of a triangulation of the sub-polygon $< i-1, i, \ldots, j >$. Provide a dynamic-programming recursive formula for $t(i,j)$. Use the recursive formula along with an appropriate matrix to determine a triangulation of the pentagon that yields the minimum cost, where the cost is the sum of the costs of each side together with the diagonals used. Remember not to overcount any of the diagonals. Draw the optimal triangulation.

40. Let $k_1, k_2, \ldots, k_n$ be keys that are to be stored in a binary-search tree. Moreover, assume that once the keys are placed in the tree, k_i will be accessed with probability p_i, for all $1 \le i \le n$. The average access cost (ACC) for the tree is defined as

$$ACC = \sum_{i=1}^{n} p_i d_i,$$

where d_i is the depth of key k_i in the tree. Note, the root of a binary tree has depth 0, while its children have depth 1, etc.. If key values $3, 1, 5, 6, 8$ are inserted into an initially empty tree (in that order), draw the tree and compute its ACC, assuming the respective probabilities are $.2, .1, .25, .2, .25$.

41. Find a binary Huffman code for the source $\mathcal{X} = \{1,2,3,4,5\}$ with respective probabilities $(.3, .2, .2, .15, .15)$. Write expressions for the average length and entropy of the code.

42. Given two strings X and Y from some alphabet, define $c(i,j)$ to be the length of the longest common subsequence for the strings X_i and Y_j, where e.g. $X_i = X_1 \cdots X_i$ is the i th prefix of string X. Provide a dynamic-programming recursive formula for $c(i,j)$. Use the recursive formula along with an appropriate matrix to determine the longest common subsequence for the strings "BLOCK" and "SLICK". Make sure to fill in all relevant entries of the matrix.

43. **Scheduling with Deadlines.** The input for this problem is a set of n tasks $a_1, \ldots, a_n$. The tasks are to be executed by a single processor starting at time $t = 0$. Each task a_i requires one unit of processing time, and has an integer deadline d_i. Moreover, if the processor finishes executing a_i at time t, where $t > d_i$ then a penalty p_i is assessed. For

example, if task a_1 has a deadline of 3 and a penalty of 10, then it must be either the first, second, or third task executed; otherwise a penalty of 10 will be assessed. Describe in detail a greedy algorithm which determines a schedule that minimizes the sum of all assessed penalties. Apply your algorithm from part a to the following problem instance.

a_i	1	2	3	4	5	6	7	8	9	10	11
d_i	4	3	1	4	3	1	4	6	8	2	7
p_i	40	50	20	30	50	30	40	10	60	20	50

44. The input for this problem is a set of n tasks $a_1, \ldots, a_n$. The tasks are to be executed by a single processor starting at time $t = 0$. Task a_i has a duration d_i associated with it. On the other hand, the processor must finish as many tasks as possible within T units of time. For every task a_i that it does not complete by this time, it is assessed a penalty p_i. Provide a dynamic-programming solution for determining what tasks the processor should execute in order to minimize the total amount of assessed penalty.

45. Consider the decision problem for deciding if a simple graph has a cycle of a given length. Call the problem k-LC. An instance of k-LC is a simple graph and a positive integer k. A positive instance is a graph which has a cycle of length at least k. Define the corresponding optimization problem for k-LC. Show that $k - LC \in \mathcal{NP}$. Prove that $k - LC$ is NP-complete using one of the following known $\mathcal{NP}$-complete problems: 3SAT, Vertex Cover, 3DM, Hamilton Cycle.

46. Give a deterministic finite automata for accepting the set of binary strings that contain three consecutive zeros.

47. Give a deterministic finite automata for accepting the set of binary strings that begin with a one and which represent the binary representation of a positive integer is a multiple of 5.

48. Write a regular expression for describing the lanaguage of all binary strings that contain three consecutive zeros.

49. Write a regular expression for describing the lanaguage of all binary strings that begin with a one and which represent the binary representation of a positive integer that is a multiple of 5.