Exam in Algorithms and Advanced Data Structures (1DV516)

2022-10-30

The exam consists of 8 problems, each worth 10 points. The questions are not arranged in order of difficulty. You may answer in Swedish or English. Illegible answers are not corrected!

- Grade A: 71 - 80 points
- Grade B: 61 - 70 points
- Grade C: 54 - 60 points
- Grade D: 47 - 53 points
- Grade E: 40 - 46 points

When an algorithm is asked for, it should be understood that it should be as efficient as possible, and it should be expressed in such a way that it is understandable.

Problem 1 (2 + 1 + 3 + 2 + 2 = 10 p)

1. Show the result of inserting 3, 1, 4, 6, 9, 2, 5, and 7 into an initially empty binary search tree. Then, show the result of deleting the root.
2. What is the minimum and maximum number of nodes at depth d in a binary search tree. The root is at depth 0. Motivate your answer.
3. Where can the smallest element reside in a max-heap with n distinct elements? And where can the largest element reside? Give both the location in the array and the location in the implicit tree structure.
4. Show that $O(N^3)$ means the same as $O(2N^3)$. Remember that a function $f(N)$ is $O(g(N))$ if $\exists c, n_0 > 0 | 0 \leq f(N) \leq cg(N), \forall n \leq n_0$
5. An algorithm takes 0.5 ms for input size 100. How long will it take for input size 500 if the running time is the following (assume low-order terms are negligible): linear, $O(N \log N)$, quadratic, and cubic?

Problem 2 (2 + 2 + 2 + 2 + 2 = 10 p)

Indicate whether the following statements are *true* or *false*. Motivate your answers. Incorrect answers are worth -1 point (but you cannot get less than 0 points for the question).

1. If a dynamic-programming problem satisfies the optimal-substructure property, then a locally optimal solution is globally optimal.
2. Every binary search tree of n nodes has height $O(logN)$.
3. Given a graph $G = (V, E)$ with cost on edges and a set $S \subseteq V$, let (u, v) be an edge such that (u, v) is the minimum cost edge between any vertex in S and vertex in $V - S$. Then the minimum spanning tree of G must include the edge (u, v).
4. Let T be a minimum spanning tree of G. Then, for any pair of vertices s and t, the shortest path from s to t in G is the path from s to t in T.
5. If a problem L_1 is polynomial time reducible to a problem L_2 and L_2 has a polynomial time algorithm, then L_1 has a polynomial time algorithm.

Problem 3 (6 + 4 = 10 p)

1. Assuming that we can use only 4 types of coins with values 1, 5, 8 and 10, given array $Coins = [coin_1, \ldots, coin_C] = [1, 5, 8, 10]$ and the amount of money to return, M, write an algorithm that computes the minimum number of coins that are necessary and executes in $O(M \cdot C)$ time. You do not need to compute the concrete number of coins of each type; just compute the total number of coins. For instance, for M=16, you should return the value 2 (two coins of value 8).
2. Given the same information $(M, [coin_1, \ldots, coin_C])$ and the decision problem, "Is there a way in which I can give the change using less than S coins?", to which class can you say that the decision problem belongs? (P, NP, NP-complete, NP-Hard). You must motivate your answer.

Problem 4 (10 p)

A toothpaste company wants to conduct some statistical studies about the monthly amount of money people spend on toothpaste. They know nobody has spent more than 1,000 Swedish kronor (SEK) during a month. They have collected data and now have an array of the monthly expenses in toothpaste of P people during the last M months. Therefore, the array contains $N = P \cdot M$ elements. They want to calculate the median value of the elements in the array. Give a solution that calculates the median expenses in toothpaste in $O(N)$. You must motivate that your solution works and argue that your solution executes in $O(N)$.

Problem 5 (2 + 2 + 2 + 2 + 2 = 10 p)

We want to represent the results of hockey games as a directed weighted graph $G(V, E)$. A game is between two teams, and the final score is based on how many goals each team scored. There are 14 teams in the league. In this simplified version, each team will play every other team once, so each team plays 13 games. We represent each team as a vertex and the final score in a game between teams v and u as edges. An edge, (u, v), with weight w means that team u scored w goals against team v. If two teams have not played each other, there are no edges between them. For a graph with 14 vertices:

1. How many edges will there be when all teams have played against each other?
2. What is the time complexity for finding the sum of the total points *scored by* a given team using adjacency lists? And using an adjacency matrix?
3. What is the time complexity for finding the sum of the total points *scored against* a given team using adjacency lists? And using an adjacency matrix?
4. What is the time complexity for finding the number of games won by a given team using adjacency lists? And using an adjacency matrix? (Note: An event between two teams u and v is won by u if the weight of (u, v) is greater than the weight of (v, u)).
5. Which representation would you choose, given the three operations above? (Adjacency lists or adjacency matrix). Motivate your answer.

Problem 6 (2 + 3 + 2 + 3 = 10 p)

Kruskal's algorithm for computing a minimum spanning tree (MST) for a weighted connected graph starts with an empty tree and repeatedly adds the remaining edge of the smallest weight, which does not introduce a cycle, to the tree. An alternative algorithm starts with the original graph and repeatedly removes an edge of the largest weight, which does not disconnect the graph.

1. Describe the algorithm in pseudo-code.
2. Give the time complexity of the algorithm. Motivate!
3. Argue that the algorithm produces a spanning tree.
4. Argue that the algorithm produces a minimum spanning tree.

Problem 7 (2 + 2 + 3 + 3 = 10 p)

1. Show that seven comparisons are required to sort five elements using any comparison-based algorithm.
2. Show the result of the following sequence of instructions in a disjoint set: $union(1, 2)$, $union(3, 4)$, $union(3, 5)$, $union(1, 7)$, $union(3, 6)$, $union(8, 9)$, $union(1, 8)$, $union(3, 10)$, $union(3, 11)$, $union(3, 12)$, $union(3, 13)$, $union(14, 15)$, $union(16, 0)$, $union(14, 16)$, $union(1, 3)$, and $union(1, 14)$.

3. What is a hash table load factor, and how do you determine it? Explain how the load factor affects lookup time. What load factor would you recommend, and why?
4. Name two collision resolution strategies for hash tables. What are the advantages and disadvantages of each?

Problem 8 (5 + 5 = 10 p)

1. Given a weighted, directed graph $G = (V, E, w)$ and the shortest path distances $d(s, u)$ from a source node s to every other node u in G. However, you are not given the actual paths. Give an algorithm that computes the shortest path from s to an input node t in $O(|V| + |E|)$ time. Argue why your algorithm is correct.
2. Given a weighted, directed graph $G = (V, E, w)$ with nonnegative-weight cycles. The graph's diameter is the maximum-weight shortest path between any two nodes. Give a polynomial-time algorithm for finding the diameter of the graph. What is the running time of your algorithm? Does it matter whether the graph is sparse or dense? Motivate!