

長庚大學 113 學年度第二學期資工所博士班演算法資格考

1. Please write down your student ID and name on the answer sheet.
 2. Please indicate the number of each your answer that is relative to the problem.
 3. Any form of cheating will lead to fail.
-

Total score of this exam is 100. Maximum deduction of 20 points for each problem that your answer.

1. An array $A[1\dots n]$ contains n distinct integers. After executing the BuildMaxHeap procedure on this array, the following array is obtained:

$A=[84, 76, 81, 65, 42, 70, 60, 23, 41, 10]$

- a. (5 points) Draw the binary max-heap as a tree that corresponds to this array.
 - b. (5 points) If we perform the ExtractMax operation (remove the maximum element and restore the heap property) once on this heap, what will be the resulting array?
 - c. (5 points) Explain why the time complexity of building a max-heap from an arbitrary array of n elements is $O(n)$ rather than $O(n \log n)$.
 - d. (5 points) Consider a min-heap instead of a max-heap. If we want to find the k -th smallest element in the array using a min-heap, describe an algorithm to accomplish this and analyze its time complexity.
2. The rod cutting problem is a classic dynamic programming problem. You are given a rod of length n inches and a table of prices $p[i]$ for $i = 1, 2, \dots, n$, where $p[i]$ is the price of a rod of length i inches. The goal is to determine the maximum revenue $r[n]$ obtainable by cutting up the rod and selling the pieces. Consider the following price table for rod lengths up to 12 inches:

Length i (inches)	1	2	3	4	5	6	7	8	9	10	11	12
Price $p[i]$ (\$)	1	5	8	9	10	17	17	20	24	30	32	35

- a. (5 points) Write down the recursive formula for calculating $r[n]$, the maximum revenue obtainable from a rod of length n .
 - b. (5 points) Using the top-down approach (memoization), calculate $r[8]$ step by step. Show your work clearly, including the subproblems you solve and the order in which you solve them.
 - c. (5 points) Implement the bottom-up approach for the rod cutting problem. Write pseudocode for a function BOTTOM-UP-ROD-CUT(p, n) that takes as input an array $p[1\dots n]$ of prices and an integer n , and returns the maximum revenue $r[n]$ and the optimal cutting solution.
 - d. (5 points) Consider a modified version of the rod cutting problem: In addition to the price table, there is a fixed cost c for each cut made. For example, if $c = \$2$ and you cut a rod of length 10 into pieces of lengths 4, 2, and 4, then you incur a cutting cost of $2 \times \$2 = \4 , which must be subtracted from the revenue. How does this modification affect the original dynamic programming solution? Provide the modified recursive formula and explain how you would adapt your algorithm to solve this modified problem.
3. The Longest Common Subsequence (LCS) problem is to find the longest subsequence common to two sequences. A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.
 - a. (10 points) Given two strings $X = \text{"ALGORITHM"}$ and $Y = \text{"LOGARITHM"}$, write down the LCS table (dynamic programming table) and determine the length of the longest common subsequence.
 - b. (10 points) Based on your LCS table from part (a), list all the longest common subsequences between X and Y .
 4. Consider the following undirected weighted graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Each edge $e \in E$ has a weight $w(e)$. Edge weights: $w(A, B) = 5$, $w(B, C) = 2$, $w(A, C) = 3$, $w(A, D) = 7$, $w(A, E) = 1$, $w(C, E) = 4$, $w(D, E) = 6$, $w(E, F) = 9$.
 - a. (8 points) Apply Kruskal's algorithm to find a minimum spanning tree (MST) for the graph. Show the edges in the order they are added to the MST and the final MST. What is the total weight of the MST?
 - b. (8 points) Apply Prim's algorithm to find an MST for the graph, starting from vertex A . For each step, show the current set of vertices in the MST, the crossing edges being considered, and the edge selected. Verify that you get the same total weight as in part (a).
 - c. (4 points) Suppose edge $w(A, E)$ is increased from 1 to 10. Without running the full MST algorithm again, determine if the MST would change. If it changes, find the new MST and its total weight.

5. Consider the following network flow graph $G = (V, E)$, where each edge (u, v) has a capacity $c(u, v)$. The source node is s , and the sink node is t . Edge capacities are the followings: $c(s, A) = 10$, $c(s, B) = 4$, $c(s, C) = 6$, $c(A, t) = 9$, $c(A, B) = 3$, $c(A, C) = 8$, $c(B, D) = 6$, $c(C, t) = 10$, $c(D, t) = 8$.
- a. (10 points) Apply the Ford-Fulkerson algorithm to find the maximum flow from s to t . Show each augmenting path that you find and the corresponding residual graph after each augmentation. Provide the final maximum flow value and indicate the flow on each edge in the final flow assignment.
- b. (10 points) Based on your maximum flow solution from part a), find a minimum cut (S, T) of the graph. List all vertices in set S and set T , and identify all edges that cross from S to T . Calculate the capacity of this cut and verify that it equals the maximum flow value (as guaranteed by the Max-Flow Min-Cut Theorem).