

長庚大學 114 學年度第一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.
-

****Please select five problems to answer. Total score of this exam is 100.
Maximum deduction of 20 points for each problem that your answer.**

1. Dynamic Programming – Subset Sum Problem

Given a set of n positive integers and a target value T , design a dynamic programming algorithm to determine whether there exists a subset whose elements sum exactly to T .

Problem: (1) Clearly define the state and recurrence relation of your algorithm. (2) Analyze the time complexity and space complexity of your solution.

2. Minimum Spanning Tree – Cut Property

Let $G = (V, E)$ be a connected, undirected graph with a real-valued weight function w defined on E . Let $A \subseteq E$ be a subset of edges that is included in some minimum spanning tree (MST) of G . Consider any cut $(S, V - S)$ of G that respects A (i.e., no edge in A crosses the cut). Let (u, v) be a light edge crossing this cut.

Problem: Prove that the edge (u, v) is safe for A ; that is, there exists an MST of G that contains $A \cup \{(u, v)\}$.

3. Quicksort – Best Case Complexity

Quicksort is a divide-and-conquer algorithm that sorts an array by partitioning it around a pivot element.

Problem: Define the *best case* scenario for Quicksort. Formulate the recurrence relation for the running time in this case, and prove that the best-case time complexity of Quicksort is $O(n \log n)$.

4. Navigation System

A city navigation system represents the road network as a directed weighted graph $G = (V, E)$. Each vertex represents an intersection. Each edge (u, v) represents a road segment from u to v , with weight $w(u, v)$ equal to the travel time.

Given a starting location s and a destination t , design an algorithm to compute the fastest route (i.e., the shortest path in terms of travel time) from s to t . Clearly define the algorithm and analyze its running time.

5. Activity Selection Problem

Suppose we are given a set of n activities: $A = \{a_1, a_2, \dots, a_n\}$, where each activity a_i has a start time s_i and a finish time f_i , with $0 \leq s_i < f_i$. Two activities a_i and a_j are compatible if their time intervals do not overlap, i.e., $f_i \leq s_j$ or $f_j \leq s_i$. The goal is to find the maximum-size subset $A' \subseteq A$ such that all activities in A' are mutually compatible.

Problem: (1) Describe in detail how to construct an optimal solution using a greedy algorithm. Clearly state the selection strategy (in terms of s_i and f_i) and outline the steps of the algorithm. (2) Prove rigorously that the greedy algorithm in Part (1) always yields an optimal solution.

6. Hash Concept

Consider a hash table with m buckets, where each bucket can store exactly one element. Assume the hash function can be computed in $O(1)$ time. When collisions

are handled using open addressing, analyze the time complexity of inserting a new key in terms of Best Case, Worst Case, and Average Case. Provide your answer with reasoning.

7. Bellman-Ford algorithm

Please describe the **Bellman-Ford algorithm** for solving the **single-source shortest path** problem in a weighted directed graph.

Problem: If the input graph G contains a **negative-weight cycle**, will the Bellman-Ford algorithm still work correctly? Provide a detailed explanation of your answer.