CS 344: Design and Analysis of Computer Algorithms

Instructor: Kangning Wang

Homework 3: Due on **March 14, 2025**

Acknowledgment: Our team members are Alexander Lin (al1655), Pranav Tikkawar (pt422), and Ivan Zheng (iz72) .

Problem 1

We are given an $n \times n$ matrix, where each entry a_{ij} is a number. We start from the top left corner, and in each step, we move one square either downward or to the right. We end when we reach the bottom right corner. We aim to maximize the sum of the numbers that we have visited.

1. Give an $O(n^2)$-time DP algorithm for the problem.
2. Now suppose that we do this process twice, but each number is only counted once even if we have visited it twice. Give an $O(n^3)$-time DP algorithm for the new problem.

Solution. Here is my solution.

Problem 2

We are given two number sequences $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$. Among their common subsequences (not necessarily consecutive), find one with the maximum sum. Give an $O(n^2)$-time DP algorithm for this problem.

Solution. Here is my solution.

Problem 3

We have n identical magic eggs. We can drop a magic egg from a building with m floors. There is an unknown threshold $t \in \{1, 2, \ldots, m\}$, so that dropping a magic egg from floor $t-1$ has no impact on the egg, but dropping a magic egg from floor t breaks it and makes it disappear.

We want to know the answer to the following question. What is the minimum number $f(n, m)$ so that we can always deduce the threshold t with at most $f(n, m)$ egg drops? For example, if $n = 1$, then the optimal strategy is to drop the egg from floor 1, then floor 2, and so on, until the egg survives floor $m-1$ or breaks. Therefore, $f(1, m) = m - 1$.

Give a DP algorithm for this problem, and write down its running time.

Solution. Here is my solution.

Problem 4

We need to build a binary search tree with n nodes whose keys are $1, 2, \ldots, n$ in this order. Let p_i be the frequency with which node i will be queried. Each time node i is queried, we need to spend time equal to one plus the depth of node i. Give an $O(n^3)$-time (or better) DP algorithm to find an optimal binary search tree that minimizes the total time spent on all queries.

Solution. Here is my solution.

Problem 5

We covered the $0/1$ knapsack and unbounded knapsack problems in class. The bounded knapsack problem is another variant. In this problem, there are n types of items, and type i has c_i identical copies. An item of type i has an integer weight $w_i \geq 1$ and a value v_i. Given a weight limit W, the goal is to find the maximum sum of values using a subset of items whose sum of weights is at most W.

Give an $O(nW^2)$-time DP algorithm for the bounded knapsack problem. As a voluntary challenge, improve the running time to $O(nW \log W)$ or $O(nW)$. (Hint: One can use the monotonic queue to achieve an $O(nW)$ running time. Alternatively, for an $O(nW \log W)$ running time, consider the binary representation of each c_i.)

Solution. Here is my solution.