

Assignment 3

COMP3121/9101 21T3

Released October 13, due November 3

In this assignment we apply dynamic programming. There are *five problems*, for a total of 100 marks.

Your solutions must be typed, machine readable PDF files. *All submissions will be checked for plagiarism!*

For each question requiring you to design a dynamic programming algorithm, you *must* specify:

- the subproblems that need to be solved,
- the recurrence relating these subproblems,
- how the final answer is obtained, and
- the time complexity of the algorithm.

For each question requiring you to design an algorithm, you *must* justify the correctness of your algorithm. If a time bound is specified in the question, you also *must* argue that your algorithm meets this time bound.

Partial credit will be awarded for progress towards a solution.

1. You are given a triangular grid of non-negative integers. The grid consists of n rows, the i th of which has i many entries. For $1 \leq j \leq i \leq n$, the j th entry in row i is denoted $T(i, j)$.

Define a *route* to be any path that starts at the top entry and ends at any entry of the bottom row, with each step going either diagonally down to the left or diagonally down to the right. Your task is to find the largest sum of numbers that can be encountered on a route.

- (a) (6 points) Consider the following greedy algorithm which attempts to construct an optimal route.

Start at the top entry. When you reach an entry $T(i, j)$, there will be two entries immediately below it; $T(i+1, j)$ to the left and $T(i+1, j+1)$ to the right. Step down to row $i+1$ in the direction of the larger of these two values.

Construct an example for which this algorithm does *not* produce the correct answer. You must include the triangle of numbers, the answer produced by this algorithm and the correct answer.

- (b) (14 points) Design a dynamic programming algorithm which solves this problem and runs in $O(n^2)$ time.
2. (20 points) You are given a non-negative integer m and an array A of length n , where each element $A[i]$ is a positive integer. Your task is to find the maximum sum of a subset S whose sum does not exceed m .
- Design a dynamic programming algorithm which solves this problem and runs in $O(mn)$ time.
3. (20 points) You are given a positive integer n and a decimal digit k . Your task is to count the number of n -digit numbers (without leading zeros) in which the digit k appears an even number of times. Note that 0 is an even number.
- Design a dynamic programming algorithm which solves this problem and runs in $O(n)$ time.
4. (20 points) You are given a non-negative integer m and an array A of length n , where each element $A[i]$ is a non-negative integer less than 2^m . Your task is to find a subarray B of maximum length such that $B[i] \& B[i + 1] \neq 0$ for every $1 \leq i < n$, where $\&$ denotes bitwise AND.
- Design a dynamic programming algorithm which solves this problem and runs in $O(mn)$ time.
5. (20 points) You are given a directed graph $G = (V, E)$, where each edge e has an associated weight $0 < w_e < 1$. You may assume that there is at least one path from every vertex u to every other vertex v , i.e. G has only one strongly connected component. Define the *safety* of a path consisting of edges e_1, e_2, \dots, e_k as $\prod_{i=1}^k w_{e_i}$. Your task is to find for each ordered pair of vertices (u, v) the maximum safety of a path from u to v .
- Design a dynamic programming algorithm which solves this problem and runs in $O(n^3)$ time.