Lab 3: Dynamic Programming

Problem 1 Coin Changing Problem

Problem Description

Suppose you are given a coin set C consists of k different values $V = \{v_1, v_2 \ldots v_k\}$.

- Each v_i $(1 \le i \le k)$ is a positive integer,
- $v_1 \le v_2 \le v_3 \le \ldots \le v_k$
- $v_1 = 1$ dollar

The problem is to pick the minimum number of coins in C to make change for n dollars. That is to find the result set $A = \{a_1, a_2 \ldots a_k\}$, such that

- $v_1 * a_1 + v_2 * a_2 + \ldots + v_k * a_k = n$
- Let $P(n) = a_1 + a_2 + \ldots + a_k$ be minimum number of coins for solving the problem.

Question

a) Define the base case of $P(n)$.

b) Suppose that the optimal solution contains at least one coin of value v_i. Write a recurrence equation for $P(n)$.
(Hint: $P(n) = ??? + P(???)$)

c) In fact, we don't know which coin appears in the optimal solution. What are the possible choices of v_i ?

1, 3, 6, 7, 9	1, 2, 5, 10
13	13

Problem 2 Longest common subsequence problem

In class, we have learnt an algorithm to solve the longest common subsequence (LCS) problem, refer to Algorithm 2.

Algorithm 2: LCS-Length($X[1..m], Y[1..n]$)

```
1  create array c[0..m][0..n]
2  for i ← 0 to n do
3      c[i, 0] ← 0
4  for j ← 0 to n do
5      c[0, j] ← 0
6  for i ← 1 to m do
7      for j ← 1 to n do
8          if X[i] = Y[j] then
9              c[i, j] ← c[i − 1, j − 1] + 1
10         else
11             c[i, j] ← max{c[i, j − 1], c[i − 1, j]}
```

1

Let $X = \langle 1, 0, 0, 1, 0, 1, 0, 1 \rangle$ and $Y = \langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$, determine a LCS of X and Y. Show your steps clearly (e.g., the table, the length of LCS and the actual LCS).

Problem 3 Checkerboard Problem

Problem Description

Suppose that you are given an $n \times n$ checkerboard and a checker. You must move the checker from the bottom edge of the board to the top edge of the board according to the following rules. At each step, you may move the checker to one of three squares:

(i) the square immediately above.

(ii) the square at diagonal left-above (but only if the checker is not already in the leftmost column).

(iii) the square at diagonal right-above (but only if the checker is not already in the rightmost column).

When you enter a square (x, y) where x and y represent the X and Y coordinates of the square, you collect the amount $p(x, y)$. Note that $p(x, y)$ is positive. Our goal is to move the checker from some square $(?, 1)$ along the bottom edge to some square $(?, n)$ along the top edge while gathering the maximum amount.

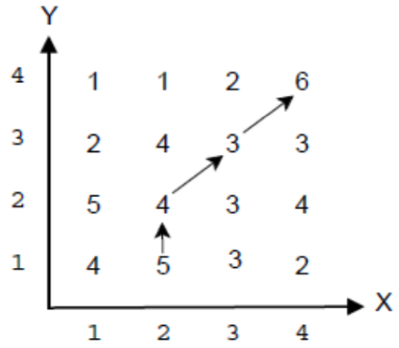

Figure 1: Checker Board Example

Example
Figure 1 shows a 4×4 checker board. The number at a square (x, y), i.e., $p(x, y)$, means the amount you can collect at that square. The arrows show the best way to move from the bottom edge to the top edge: $(2, 1) \rightarrow (2, 2) \rightarrow (3, 3) \rightarrow (4, 4)$. Here, we gather the amount: $5 + 4 + 3 + 6 = 18$.

Questions

(a) Let $c(x, y)$ be the maximum amount gathered by moving the checker from some square $(?, 1)$ along the bottom edge to the square (x, y). Define the base case of $c(x, y)$ and give a recurrence of $c(x, y)$.

(b) Design an algorithm that figures out the set of moves that will move the checker from some square $(?, 1)$ along the bottom edge to some square $(?, n)$ along the top edge while gathering the maximum amount.

(c) Show your running steps of your algorithm on the example of Figure 1.