COMP 6651: Worksheet 4

Dynamic Programming

Note: you should always try to give as efficient algorithm as possible.

For each of the problems below that ask to design a dynamic programming algorithm, you should follow the following template:

- Describe the semantic array. State where the solution is stored in the dynamic programming table.
- Describe the computational array. Do not forget the base case.
- Briefly argue why the above two arrays are equivalent.
- Write your algorithm in pseudocode. It should be an iterative implementation (no recursion, memoization).
- Justify the runtime of your dynamic programming solution.

1. (Maximum Subarray Problem revisited) You are given an array $A[1..n]$ of numbers. Design a dynamic programming algorithm that finds the largest sum of elements in a contiguous subarray $A[i..j]$. Your algorithm should run in time $O(n)$ assuming that all operations on numbers take $O(1)$ time.

2. A sequence $X[1..n]$ of numbers is oscillating if $X[i] < X[i+1]$ for all even i and $X[i] > X[i+1]$ for all odd i. Design a dynamic programming algorithm that finds the length of the longest oscillating subsequence of an arbitrary array A of integers.

3. A sequence $X[1..n]$ of numbers is convex if $2X[i] < X[i-1] + X[i+1]$ for all $i \in \{2, \dots, n-1\}$. Design a dynamic programming algorithm that finds the length of the longest convex subsequence of an arbitrary array A of integers.

4. Describe a dynamic programming algorithm to compute the number of times that one given array $X[1..k]$ appears as a subsequence of another given array $Y[1..n]$.

5. Suppose we are given a set L of n line segments in the plane where each segment has one endpoint on the line $y = 0$ and one endpoint on the line $y = 1$, and all $2n$ endpoints are distinct. Describe and analyze a dynamic programming algorithm to compute the largest subset of L in which no pair of segments intersect.

6. You are opening a series of restaurants along Highway 401. The n possible locations are along a straight line, and the distances of these locations from the start of Highway 401 are $m_1, m_2, \dots, m_n$ (in kilometers, in increasing order). The constraints are as follows:

   - at each location, you may open at most one restaurant. The expected profit from opening a restaurant at location i is p_i, where $p_i > 0$;
   - any two restaurants should be at least k kilometers apart, where k is a given positive integer.

   Give an efficient dynamic programming algorithm to compute the maximum obtainable expected profit subject to given constraints.

7. A subsequence is palindromic if it reads the same forward as backward. For example, consider the following sequence of ASCII characters:

$$C, O, M, P, 6, 6, 5, 1, I, S, P, I, E, C, E, O, F, C, A, K, E$$

The above sequence has many palindromic subsequences, including $6, 6$ and E, C, E. Another example of a palindromic subsequence is $C, P, 6, 6, P, C$. Example of a non-palindromic subsequence is $C, 6, 6, P, C$. Design a dynamic programming algorithm that takes a string of n ASCII characters $x[1..n]$ and computes the length of the longest palindromic subsequence. Your algorithm should run in $O(n^2)$ time.

8. You are given a string of n characters $s[1..n]$, which you believe to be a corrupted text document in which all punctuation has vanished (so that it looks something like "ireallylikedynamicprogrammingproblems..."). You wish to reconstruct the document using a dictionary, which is available in the form of a Boolean function $\text{dict}(\cdot)$: for any string w,

$$\text{dict}(w) = \begin{cases} \text{true}, & \text{if } w \text{ is a valid word} \\ \text{false}, & \text{otherwise.} \end{cases}$$

   Give an efficient dynamic programming algorithm that determines whether the string $s[1..n]$ can be reconstructed as a sequence of valid words.

9. You are a manufacturer of chocolate goods. You start with a large rectangular chocolate bar that consists of $W \times H$ tiles arranged into W rows and H columns. You have a machine that can make a horizontal or a vertical break along tile edges. There are n possible goods that you can manufacture. Good i is described by w_i, h_i and p_i. This means that manufacturing good i requires a rectangular chocolate bar consisting of exactly $w_i \times h_i$ tiles (w_i rows and h_i columns). The value of p_i is how much profit you can make from good i. You can manufacture the same good more than once. What is the maximum overall profit that you can get starting with the large rectangular chocolate bar, performing a series of break operations, and manufacturing goods? Design an efficient dynamic programming algorithm. Note: W, H, w_i, h_i are positive integers and the p_i are reals.

10. Suppose you are given a sequence of integers separated by $+$ and $-$ signs, e.g.:

$$12 + 34 - 10 + 5 - 3 - 7$$

    You can change the value by adding parentheses in different places, e.g.:

$$(((12 + 34) - 10) + (5 - (3 - 7))) = 45$$

$$((((12 + 34) - (10 + 5)) - 3) - 7) = 21$$

    Describe and analyze an algorithm to compute, given a list of integers separated by $+$ and $-$ signs, the maximum possible value the expression can take by adding parentheses.

    You may only use parentheses to group additions and subtractions; you are not allowed to create multiplication like $5 + 3(-2)(-1) = 11$.

11. You are given a convex polygon P on n vertices in the plane. A triangulation of P is a collection of $n - 3$ diagonals of P such that no two diagonals intersect except possibly at their endpoints. The cost of triangulation is the sum of the lengths of the diagonals in it. Design an efficient dynamic programming algorithm for finding a triangulation of minimum cost.

12. Suppose that two teams, A and B, are playing a match to see who is the first to win n games. Suppose that for each particular game each team has 50% chance of winning it, independent of what happened in the past. Suppose that the teams have already played $i + j$ games, of which A won i and B won j. Give an efficient dynamic programming algorithm to compute the probability that A will go on to win the match.

13. Given an unlimited supply of coins of denominations $x_1, \ldots, x_n$, we wish to make change for a value v; that is, we wish to find a set of coins whose total value is v. Notice that this might not be possible. Give an $O(nv)$ dynamic programming algorithm for the following problem:

    Input: $x_1, x_2, \ldots, x_n; v$.

    Output: is it possible to make change for v using denominations $x_1, \ldots, x_n$?

14. Given an unlimited supply of coins of denominations $x_1, \ldots, x_n$, we wish to make change for a value v using at most k coins; that is, we wish to find a set of $\leq k$ coins whose total value is v. Notice that this might not be possible. Give a dynamic programming algorithm for the following problem:

    Input: $x_1, x_2, \ldots, x_n; v$.

    Output: is it possible to make change for v using at most k coins of denominations $x_1, \ldots, x_n$?

15. Give an $O(nt)$ algorithm for the following task:

    Input: $A[1..n]$ – an array of n positive integers; t – a positive integer.

    Output: does some subset of $A[1..n]$ add up to exactly t?

16. Consider the 3-Partition problem: given an array $A[1..n]$ of n positive integers, your goal is to determine if it is possible to partition $\{1, 2, \ldots, n\}$ into three disjoint subsets I, J, K such that

$$\sum_{i \in I} A[i] = \sum_{j \in J} A[j] = \sum_{k \in K} A[k] = \frac{1}{3} \sum_{i=1}^{n} A[i].$$

    Devise a dynamic programming algorithm for 3-Partition that runs in polynomial time in n and $\sum_{i=1}^{n} A[i]$.