Exercises - Dynamic Programming 2 - Algoritmiek

Tutorial February 15, 2024

1. **LCS:** Consider the dynamic programming algorithm for LCS discussed during the lecture.

   (a) Show how to find a solution (so a longest common subsequence) of two strings without using the helper table b.

   (b) Use your dynamic programming algorithm to find a solution for $X = \langle 1, 0, 0, 1, 0, 1, 0, 1 \rangle$ and $Y = \langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$.
   Compare your answer and the table using the following visualization:
   `https://www.cs.usfca.edu/~galles/visualization/DPLCS.html`

2. **The Poor Pilgrim:** A pilgrim goes to Rome on foot. The distance is K km. On a day, the pilgrim can walk at most L km. Fortunately, there are $n + 1$ lodges on the pilgrim's way. Lodge i lies at distance d_i from the start of the pilgrim's journey; assume $d_0 \leq \cdots \leq d_n$, where d_0 is the start and $d_n = K$ is in Rome. The distance between any two lodges is at most L. Spending the night at lodge i costs the pilgrim p_i gold coins.

   (a) Give an algorithm that computes the smallest number of gold coins the pilgrim needs to complete his journey. Argue that your algorithm is correct and analyze its running time.

   (b) Suppose the pilgrim has B gold coins. Give an algorithm that computes in $O(Bn^2)$ time how the pilgrim can reach Rome by spending as few nights in lodges as possible, while not spending more than B gold coins.

3. **Snapshots:** A city has n famous attractions, $A_1, \ldots, A_n$. You are at your AirHotel, A_0. As a photographer, you want to snap snapshots of the attractions. A snapshot of A_i has value g_i. However, you also have limited time T before darkness sets in and you need to be back at your AirHotel. It takes $d[i, j]$ time to travel from A_i to A_j. Use dynamic programming to decide on a tour that yields snapshots of highest value. Analyze the running time of your algorithm

4. **Dicy race:** Bert and Ernie are playing a simple board game. They start at place 1 on a board with n places. Bert starts. Taking turns alternatingly, they throw a regular six-sided dice. They move forward as many places as the number they throw with the dice. Who ends up at the last place, or passes it, wins the game.
   Because Bert starts, he has a probability more than $\frac{1}{2}$ to win the game. Given a value of n, we want to compute this probability. We will do this using dynamic programming.
   Suppose $B(i, j)$ is the probability that Bert wins if Bert still has i steps until the end and Ernie still has j steps until the end, and it's Bert's turn.
   Suppose $E(i, j)$ is the probability that Ernie wins if Bert still has i steps until the end and Ernie still has j steps until the end, and it's Ernie's turn.

   (a) Why do we want to know $B(n - 1, n - 1)$?

   (b) Give recurrences for $B(i, j)$ and $E(i, j)$ if $i \geq 6$.

   (c) Give the base cases for the recurrence: $B(1, j)$, $B(2, j)$, $B(3, j)$, $B(4, j)$, $B(5, j)$.

   (d) What is the order of computation for the recurrence?

   (e) Give pseudocode for an algorithm that computes $B(n - 1, n - 1)$ in $O(n^2)$ time.

5. **Party!:** You are the organizer of a party at the company for which you intern and need to send out invitations. The company has a strict hierarchy: every employee has a single supervisor, who also has a supervisor, etc., right until the CEO. Every supervisor can have multiple direct subordinates (who again have multiple direct subordinates etc.). Partying with your supervisor or your direct subordinates is always a bit weird; therefore, if you invite an employee to the party, you are not allowed to invite their supervisor nor their direct subordinates. Still, you want to invite as many employees to the party as possible.

   (a) Give a dynamic programming algorithm for this problem. Use the steps discussed in class!

   (b) Analyze the running time of your algorithm.

   (c) The company has given each employee a party-ranking. The higher the total rank of the invited employees, the more fun the party will be. Adapt your algorithm to arrange the best party evah!