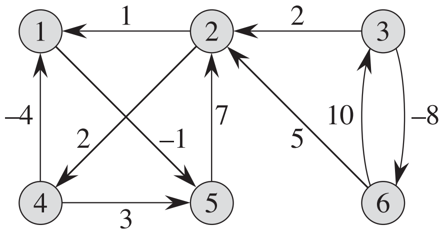
Weekly Assignment 10

November 19, 2019

Exercise 1. *Weight: 20%*

Run the Floyd-Warshall algorithm on the following graph. Show $D^{(0)}$ and the matrix $D^{(k)}$ after each iteration.



Exercise 2. *Weight: 20%*

Modify the Floyd-Warshall algorithm to compute the matrices $\Pi^{(k)}$ of predecessors (see lecture slides or equations (25.6) and (25.7) in Cormen et al.'s textbook). Then write an algorithm Print-Paths(Π) that prints the shortest paths between all pairs of vertices, one path per line. The path from i to j should be printed as its list of vertices $i \dots j$. If there is no such path, the line should display an error message. Suppose you have procedures **print**(x) that prints x and **newline** that starts a new line.

Exercise 3. *Weight: 10%*

Answer the following multiple choice questions:

1. Which of the following is/are property/properties of a dynamic programming problem?
   (a) Optimal substructure
   (b) Overlapping subproblems

   (c) Greedy approach
   (d) Both optimal substructure and overlapping subproblems

2. If an optimal solution can be created for a problem by constructing optimal solutions for its subproblems, the problem possesses which property?
   (a) Overlapping subproblems
   (b) Optimal substructure
   (c) Memoization
   (d) Greedy
3. If a problem can be broken into subproblems which are reused several times, the problem possesses which property?
   (a) Overlapping subproblems
   (b) Optimal substructure
   (c) Memoization
   (d) Greedy
4. If a problem can be solved by combining optimal solutions to non-overlapping problems, the strategy is called
   (a) Dynamic programming
   (b) Greedy
   (c) Divide and conquer
   (d) Recursion
5. In dynamic programming, the technique of storing the previously calculated values is called
   (a) Saving value property
   (b) Storing value property
   (c) Memoization
   (d) Mapping

Exercise 4. *Weight: 25%*

A contiguous subsequence of a list L is a subsequence made up of consecutive elements of L. For instance, if L is $5, 15, -30, 10, -5, 40, 10$ then $15, -30, 10$ is a contiguous subsequence but $5, 15, 40$ is not. Use dynamic programming to give a linear-time algorithm for the following task:
Input: A list of numbers - $a_1, a_2, \ldots, a_n$.
Output: The contiguous subsequence of maximum sum (a subsequence of length zero has sum zero).
For the given example, the answer would be $10, -5, 40, 10$ with a sum of 55. Specify the recursion equations on which your solution is based, explain why these equations are correct, describe a bottom-up implementation based on the recursion equations, and explain why the time complexity is linear.
Hint: compute, for each index $1 \leq j \leq n$, the subsequence with the largest sum ending at a_j. Moreover, compute the subsequence with the largest sum anywhere in $a_1, \ldots, a_j$.

Exercise 5. *Weight: 25%*

Suppose you are going on a biking trip during your holiday. You start on the road at kilometer post 0. Along the way there are n hotels, at kilometer posts $h_1 < h_2 < ... < h_n$, where each h_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance h_n), which is your destination. Ideally you would like to travel 200 kilometer a day, but this may not be possible (depending on the distance between the hotels). If you travel m kilometers during a day, the penalty for that day is $(200 - m)^2$. You want to plan your trip so as to minimize the total penalty, that is, the sum, over all travel days, of the daily penalties.

Use dynamic programming to give an efficient algorithm that determines the optimal sequence of hotels at which to stop. Specify the recursion equations on which your solution is based, explain why these equations are correct, describe a bottom-up implementation based on the recursion equations, and analyze the time complexity of your algorithm.