Layla Oesper, Jeff Ondich

One question of your choice (either #1 or #2 or #3) is due by **5:00PM Tuesday, May 21** and will be graded on completion only (1 point). Your full answers to all questions are due by **5:00PM Friday, May 24**. Each solution will be graded for correctness and clarity (4 points per question). Read the course information page/syllabus for further info about this 4-point scale.

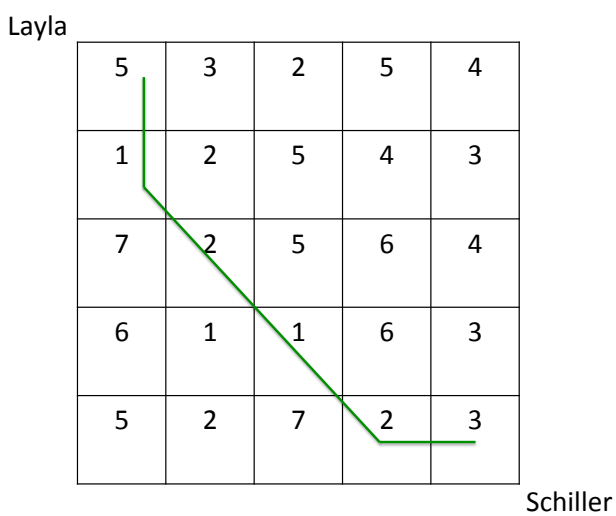
0. Estimate the amount of time you spent on each question and include it at the top of your solution. Also, list your collaborators for each question at the top of your solution.

1. Layla haven't seen Schiller in a really long time. So, she's been sending out secret messages across campus to try and get a glimpse of him - just to make sure he is doing alright. Then, just last night a carrier pigeon dropped off a note to Layla as she was walking to her car from Olin. Schiller's keepers have decided that if she proves her worth first, they may allow her to see him.

   Specifically, they have set up a grid of obstacles (literally a grid of obstacles) on the bald spot that Layla needs to pass through in order to get a glimpse of Schiller. Layla will start at the top-left corner of the grid and Schiller is waiting at the bottom right hand corner of the grid. Layla has to find a sequence of moves through the grid from her starting spot to Schiller's location. However, she is only allowed to move one spot at a time either to the right, down, or down-right diagonally. Each spot on the grid has some challenge that she has to face before she can move on. Each challenge has a different difficulty factor. So, she need to find the "cheapest route" from her starting location to Schiller, in fact if she does not take the "cheapest route" Schiller will immediately disappear! Furthermore, the keepers of Schiller have given Layla a very short time for deciding her route, so she needs to find a really fast algorithm in order to choose her route. Layla needs your help to design a dynamic programming algorithm to solve this problem.

   Formally, you are given an $n \times n$ grid containing integer values (representing the obstacle at that location). **You need to find a cheapest route** from the top left corner, or index (1,1), to the bottom right corner, or index (n,n) that at each step moves one spot down, right or down-right diagonally. Such a route is just a sequence of pairs such as (1,1) to (2,2) to (2,3) etc. Give an $O(n^2)$ dynamic programming algorithm to solve this problem. The figure below shows one such problem and the correct path to take (the path has a score of 14). Note, in some instances there may be multiple such paths, but you only need to find one.

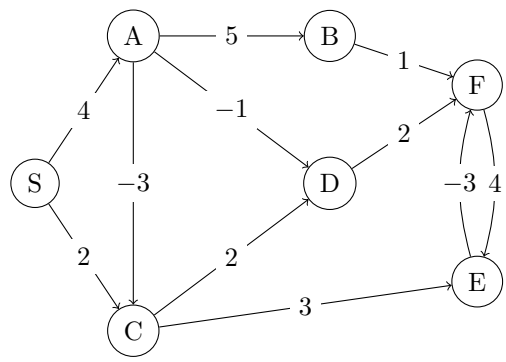
   As always, describe your algorithm, give a proof that it solves the desired problem and an analysis of its running time.



Layla Oesper, Jeff Ondich

2. (a) Give an example of a graph G with negative edge weights, but no negative length cycles, where running Dijkstra's algorithm on G from some start node s will not produce the correct shortest paths. Describe precisely what Dijkstra's algorithm will get wrong when run on this graph.

   (b) The Bellman-Ford algorithm discussed in the reading finds the length of all shortest paths to some target node t. However, you often see the algorithm used to find the length of all shortest paths from some start node s to all other nodes, as we discussed in class. Run the Bellman-Ford algorithm on the following graph to find the length of all shortest paths from node s to all other nodes. Include in your answer:(1) the final dynamic programming table M that you compute (which stores lengths of shortest paths); (2) the length of the shortest path found from s to all other nodes in the graph; and (3) the sequence of nodes on the shortest paths from s to all other nodes.



3. (a) Let $f(n) = 4n^3 + 2n + 6$ and let $g(n) = n^3$. Use the formal definition of O to prove that $f(n)$ is $O(g(n))$.

   (b) Let $f(n) = n^3 + 10n + 4$ and let $g(n) = n^4$. Use the formal definition of O to prove that $f(n)$ is $O(g(n))$.

   (c) Let $f(n) = 4n^2 + \log n$ and let $g(n) = n$. Use the formal definition of Ω to prove that $f(n)$ is $\Omega(g(n))$.

   (d) Given an example of a function $f(n)$ that is $O(n^2)$ but that is not $\Omega(n^2)$. Explain your answer.