Homework 3

Due Friday, February 15 at 5:00 PM

"My life is over. I've been a good girl... I've never lied... Except when necessary. I've always bought my parents expensive gifts... Using their credit cards, of course!"
-Sharpay Evans

Handing In

To hand in a homework, go to the directory where your work is saved and run `cs0160_handin hwX` where X is the number of the homework. Make sure that your written work is saved as a .pdf file, and any Python problems are completed in the same directory or a subdirectory. You can re-handin any work by running the handin script again. We'll only grade your most recent submission. To install stencil Python files for a homework, run `cs0160_install hwX`. **You will lose points if you do not hand in your written work as a .pdf file.**

1 Written Problems

Problem 3.1

Induction

Silly premise: In order to defeat the evil Gabriella and take her mans Troy back, Sharpay is seeking out fellow sneks to be her allies. However, she can only get her hands on *real* allies if she uses induction to guess the location of her fanbase. CS16, *We're All In This Together* to help Sharpay. Will you *Work This Out* and learn induction along the way? Help Sharpay use induction to solve the following problems and reach her goal of taking her mans Troy back from Gabriella. Its *Now Or Never* folks.

---

1. Consider the statement $P(n) : (1+a)^n \geq 1 + a(n-1)$, given some $a \geq 0$. Use induction to prove $P(n)$ for all $n \geq 1$. While it may be a helpful reference, do not use the results of the proof in the **induction handout** (on the Docs page) in this proof–the problems are similar but not the same.

2. Consider the statement $P(n) : (1+a)^n \geq 1 + a(n+1)$. Prove that $P(n-1) \implies P(n)$ for $a > 0, n > 0$. Why doesn't this prove that $P(n)$ holds for every $n > 0$?

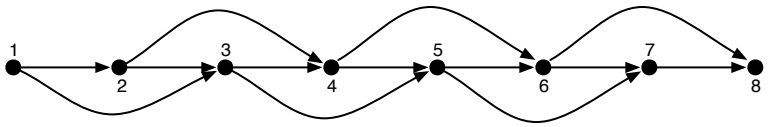

Figure 1: The input graph.

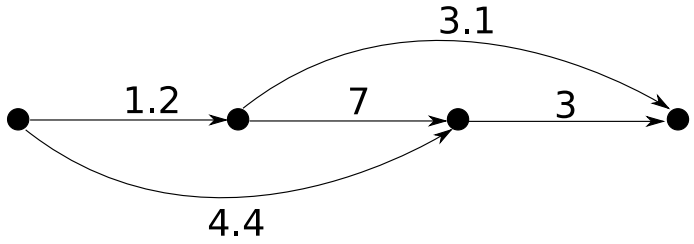

Figure 2: The input graph with weights on the edges.

Problem 3.2

Greedy vs. Dynamic Programs

Let G be a graph (a collection of nodes and directed arcs) with n nodes, labeled $1, 2, \ldots, n$, and arcs as drawn in Figure 1 for $n = 8$. There are two types of arcs:

1. For $1 \leq i \leq n-1$, there is an arc $(i, i+1)$.
2. For $1 \leq i \leq n-2$, there is an arc $(i, i+2)$.

Let $cost(i, j)$ be a function that takes as input two nodes connected by an edge in G and returns the cost of the edge between them. See Figure 2 for an example of arcs labelled with their costs. Consider the problem of finding a low-cost path from node 1 to node n (the cost of a path is the sum of the costs of the arcs on the path).

For the following problems you do not have to give detailed code. It suffices to clearly explain your algorithms in a few sentences or formulas. Your solution only needs to return the minimum cost, not the path itself.

1. Describe a greedy algorithm for solving the problem. A greedy algorithm is an algorithm that makes the best choice locally at each stage with the hope of finding a global optimum solution. Does your algorithm find the optimum for Figure 2? Will it always find the optimum? Prove or give a counterexample. (A counterexample would be a single instance of a graph in the specified form, and a set of weights for that graph (i.e., a picture like Figure 2 above), with the property that the path produced by the greedy algorithm is not optimal).

2. Describe a dynamic programming approach (like the one we used for solving seam-carving) to find an optimal solution. An algorithm that uses

dynamic programming breaks the larger problem into smaller subproblems then uses the solutions to those subproblems to build up to solving the overall problem. See the lecture slides for more information.

2 Python Problems

Problem 3.3

Bop To The Top

Sharpay and Ryan are about to perform for callbacks and need to be number one to secure the lead roles in the school musical. This is crucial in preserving Sharpay's reputation as the queen of the school.

However, as they are going through their pre-show routine, the Evans twins find out they are being sabotaged by the stage crew! The sparkly gold ladder that is essential to their performance has been constructed so its rungs are spaced unevenly. At the end of their act Sharpay needs to *Bop to the Top* of the ladder in the least number of steps possible to blow away the competition. Because of the ladder's poor construction, the number of rungs she can skip over with one step varies based on where she is on the ladder.

Despite this setback Sharpay is determined to have a perfect act and be number one. By closely examining the ladder, she's figured out the maximum number of rungs she can step over from each of the rungs. She's written this down in an array of n positive integers, where n represents the total number of rungs on the ladder. This means if the first element of the array were 3, then she can move at most 3 rungs above in one step. If the value is 0, she can not move past that rung. Write a function that finds the minimum number of steps from the first rung to the top of the ladder in order to help Sharpay *Bop to the Top*! Also, make sure your solution uses dynamic programming.

Example:

Let's say you have the array $[3, 1, 2, 0, 8]$. The minimum path would be to step 2 rungs from the start and then 2 rungs to the end. Therefore, your minimum number of steps would be 2.

Notes:

- All you need to return is the minimum number of steps. Dont worry about returning the actual path.
- The maximum number of rungs is given by an element in the array, which means a step can be made up of less rungs. For example, in the array above, the first step can skip 3, 2, or 1 rung(s). Here, the best choice would be to skip 2, rather than the maximum of 3.
- You will always start at the first rung (aka the first element in the array) and the top is reached once you reach the last rung. It is not necessary to step past the last rung.
- If the end can not be reached you should return None.

Problem 3.4

The Jewelry Set Conundrum

Silly premise: After successfully convincing Troy to come work in the Lava Springs, Sharpay realizes she is up for an intense competition to win his heart, as Troy only agreed to work there with Gabriella and his friends. Sharpay wants to steal his heart by investing in her summer fashion.

She wants to buy all the new handbags available, but her dad's account is running low, as he spent a lot of money employing Sharpay's friends. The amount of money that Mr. Evans gives her could not get her any of the designer handbag that she wants. Therefore, she thought that she could invest in selling jewelry sets, something she is really passionate and knowledgeable about. Lucky for her, the expertise in business of her dad and his business friends allow Sharpay to predict the future market value of jewelry sets and contribute to late stage capitalism. However, she has never done any real "business" before, so she needs your help!

Suppose Sharpay has 180 dollars to invest. She knows how much the value of the jewelry set will fluctuate for n days, and she has written it down in an array of n numbers $r[0] \dots r[n-1]$, where $r[i]$ represents the change in value for day i.

You are investing on behalf of Sharpay, and you can choose the days on which you buy and sell jewelry sets. You can only buy and sell them once, so you must hold onto them for a number of consecutive days. Your goal is to maximize your profits. Use Python to implement a linear time algorithm for finding the maximum amount of money you could have. Assume that jewelry sets don't lose value on their own.

Example:

Let $r = [-1, 2, 7, -8, 13, -2]$. To achieve maximum profits, you should buy on the second day ($i = 1$) and sell on the fifth day ($i = 4$). Therefore, when you sell out he will have $180 + 2 + 7 - 8 + 13 = 194$ dollars. Your function should return 194.

Notes:

- Money is gained (or lost) on both the day you buy and sell the jewelry set. For example, if you buy on the 12th day and sell on the 14th and if $r[11] = 10, r[12] = -3, r[13] = 4$ then you will end up having $180+10-3+4 = 191$ dollars.
- Your algorithm is not required to return the buying and selling days. You only need to return the maximum money that can be achieved by your investments.
- Remember that one option is to not buy at all, leaving you at 180 dollars. Another option is for you to buy and sell on the same day (your worth would be 180 + the dollar change for that day).