

## CS 170 Homework 3

Due 9/19/2022, at 10:00 pm (grace period until 11:59pm)

### 1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write “none”.

### 2 Updating Labels

You are given a tree  $T = (V, E)$  with a designated root node  $r$ , and a non-negative integer label  $l(v)$  for each node  $v$ . We wish to relabel each vertex  $v$ , such that  $l_{\text{new}}(v)$  is equal to  $l(w)$ , where  $w$  is the  $k$ -th ancestor of  $v$  in the tree for  $k = l(v)$ . In other words,  $v$  “inherits” the label from its  $k$ -th ancestor. Follow the convention that the root node,  $r$ , is its own parent.

Design a linear time algorithm to compute the new label,  $l_{\text{new}}(v)$ , for each  $v$  in  $V$ . Just describe the algorithm and give a runtime analysis in terms of  $n = |V|$  and  $m = |E|$ ; no proof of correctness is necessary.

### 3 Where’s the Graph?

Each of the following problems can be solved with techniques taught in lecture. Construct a simple directed graph and write an algorithm for each problem by black-boxing algorithms taught in lecture and in the textbook.

- (a) Sarah wants to do an extra credit problem for her math class. She is given three numbers: 1,  $x$ , and  $y$ . Starting from  $x$ , she needs to find the shortest sequence of additions, subtractions, and divisions (only possible when the number is divisible by  $y$ ) using 1 and  $y$  to get to 2022. If there are multiple sequences with the shortest length, return any one of them. She can use 1 and  $y$  multiple times. Give an algorithm that Sarah can query to get this sequence of arithmetic operations.
- (b) There are  $n$  different species of Pokemon, all descended from the original Mew. For any species of Pokemon, Professor Juniper knows all of the species *directly* descended from it. She wants to write a program that answers queries about these Pokemon. The program would have two inputs:  $a$  and  $b$ , which represent two different species of Pokemon. Her program would then output one of three options in constant time (the time complexity cannot rely on  $n$ ):
  - (1)  $a$  is descended from  $b$ .
  - (2)  $b$  is descended from  $a$ .
  - (3)  $a$  and  $b$  share a common ancestor, but neither are descended from each other.

Unfortunately, Professor Juniper’s laptop is very old and its SSD drive only has enough space to store up to  $O(n)$  pieces of data for the program. Give an algorithm that Professor Juniper’s program could use to solve the problem above given the constraints.

*Hint: Professor Juniper can run some algorithm on her data before all of her queries and store the outputs of the algorithm for her program; time taken for this precomputation is not considered in the query run time.*

- (c) Bob has  $n$  different boxes. He wants to send the famous "Blue Roses' Unicorn" figurine from his glass menagerie to his crush. To protect it, he will put it in a sequence of boxes. Each box has a weight  $w$  and size  $s$ ; with advances in technology, some boxes have negative weight. A box  $a$  inside a box  $b$  cannot be more than 15% smaller than the size of box  $b$ ; otherwise, the box will move, and the precious figurine will shatter. The figurine needs to be placed in the smallest box  $x$  of Bob's box collection.

Bob (and Bob's computer) can ask his digital home assistant Falexa to give him a list of all boxes less than 15% smaller (but not necessarily lighter) than a given box  $c$ . Bob will need to pay postage for each unit of weight. Find an algorithm that will find the lightest sequence of boxes that can fit in each other in linear time (in terms of the graph).

*Hint: how can we create a graph knowing that no larger box can fit into a smaller box, and what property does this graph have?*

## 4 The Greatest Roads in America

Arguably, one of the best things to do in America is to take a great American road trip. And in America there are some amazing roads to drive on (think Pacific Coast Highway, Route 66 etc). An intrepid traveler has chosen to set course across America in search of some amazing driving. What is the length of the shortest path that hits at least  $k$  of these amazing roads?

Assume that the roads in America can be expressed as a directed weighted graph  $G = (V, E, d)$ , and that our traveler wishes to drive across at least  $k$  roads from the subset  $R \subseteq E$  of "amazing" roads. Furthermore, assume that the traveler starts and ends at her home  $a \in V$ . You may also assume that the traveler is fine with repeating roads from  $R$ , i.e. the  $k$  roads chosen from  $R$  need not be unique.

Design an efficient algorithm to solve this problem. Provide a 3-part solution with run-time in terms of  $n = |V|$ ,  $m = |E|$ ,  $k$ .

*Hint: Create a new graph  $G'$  based on  $G$  such that for some  $s', t'$  in  $G'$ , each path from  $s'$  to  $t'$  in  $G'$  corresponds to a path of the same length from  $a$  to itself in  $G$  containing at least  $k$  roads in  $R$ . It may be easier to start by trying to solve the problem for  $k = 1$ .*

## 5 Detecting Cyclic Vertices

Design an efficient algorithm that given a directed graph  $G = (V, E)$ , outputs the set of all vertices  $v$  such that there is a cycle containing  $v$ . In other words, your algorithm should output a set containing all vertices  $v$  such that there exists a path from  $v$  to itself in  $G$  that traverses at least one other unique vertex  $u \neq v \in V$ .

Provide a 3-part solution with run-time in terms of  $n = |V|$  and  $m = |E|$ .

## 6 Coding Questions

For this week's coding questions, we'll be implementing FFT, DFS, and BFS, and applying them to common problems. The Jupyter Notebooks are located under the `hw3` folder [here](#). You can also download these files to your personal computer and complete the exercises locally. Please complete both jupyter notebooks and make sure to submit them to the right subparts on gradescope.

(a) `fft.ipynb`

(b) `dfs_bfs.ipynb`

Notes:

- *Submission Instructions:* Please merge your completed Jupyter Notebooks with your written solutions for other questions and submit one merged pdf file to Gradescope.
- *OH/HWP Instructions:* While we will be providing conceptual help on the coding portion of homeworks, OH staff will not look at your code and/or help you debug.
- *Academic Honesty Guideline:* We realize that code for some of the algorithms we ask you to implement may be readily available online, but we strongly encourage you to not directly copy code from these sources. Instead, try to refer to the resources mentioned in the notebook and come up with code yourself. That being said, we **do acknowledge** that there may not be many different ways to code up particular algorithms and that your solution may be similar to other solutions available online.