COMP 285 (NC A&T, Fall '22) Homework 5 (100 pt)

Fun with Graphs

Due 10/25/22 @ 11:59PM ET

Submitting

In order to submit this assignment, you must download your assignment as a ZIP file and upload it to Gradescope. There will be a written component and a coding component.

For the written component, you will write your answers in the corresponding `.txt` files.

For the coding component, you will write your solution in `answers.cpp`. We will use an autograder to test your solutions - the results of which can be seen after submitting to Gradescope. Note that you have unlimited tries to submit. In order to receive full credit, you must also provide documentation on your approach, see the commented sections in the `answers.cpp` file.

Note: This homework requires you to input your Banner ID as a string in the getBannerID method in `answers.cpp`.

Question 1: Undirected Graphs (15 pts)

You should fill in the `getBannerId` function with your Banner ID in `answers.cpp`. When you hit run on replit, you will receive an auto-generated undirected graph such as this example:

```
##################
### Question 1 ###
##################
Your graph is G = (V, E) with *undirected edges*, where
V = {0, 1, 2, 3, 4, 5, 6, 7} and
E = {[(0,2), (0,4), (0,5), (1,3), (2,5), (2,6), (3,6), (4,6), (4,7)]}
```

Answer the following questions with the graph provided.

a. Drawing the Unweighted Graph (3 pt) Represent your graph as a picture. I recommend using Google Drawings for this. I've provided a template that you can make a copy of and use.

Upload your picture as "undirected-graph.png" to Repl.it (it will be downloaded as part of your submission).

For example, the answer to the example graph would look like this:

b. Adjacency Matrix (3 pt): Represent your graph as an adjacency matrix (2D array of bools).

Write your solution in `getAdjacencyMatrix` in `answers.cpp`

c. Adjacency List (3 pt): Represent your graph as an adjacency list (list of list of ints).

Write your solution in `getAdjacencyList` in `answers.cpp`

d. BFS on Graphs (3 pt): Given the following algorithm, what would be the ordering of nodes visited when we call `bfs(G, 0, 6)` ? Put your answer in getBfsOrder.

```
algorithm bfs
  Input: undirected graph G = (V,E), int s and int t
  Output: true if there's a path from s to t; false otherwise

  frontier = Queue of integers
  visited = new boolean array of size |V|
  frontier.add(s)
  visited[s] = true
  while frontier.size() > 0
    currNode = frontier.remove()
    if currNode == t
      return true
    for each neighbor of currNode, starting at the smallest labeled neighbor
      if !visited[neighbor]
        visited[neighbor] = true
        frontier.add(neighbor)
  return false
```

Write your solution in `getBfsOrder` in `answers.cpp`

e. DFS on Graphs (3 pt) Given the following algorithm, what would be the ordering of nodes visited when we call `dfs(G, 0, 6)`?

```
algorithm dfs
  Input: undirected graph G = (V, E), int s and int t
  Output: true if there's a path from s to t; false otherwise

  visited = new boolean array of size |V|
  return dfsHelper(G, s, t, visited)

algorithm dfsHelper
  Input: undirected graph G = (V, E), int s and int t, and boolean array visited
  Output: true if there's a path from s to t; false otherwise

  visited[s] = true
  if s == t
    return true;
  for each neighbor of s, starting at the smallest labeled neighbor
    if !visited[neighbor] and dfsHelper(G, neighbor, t, visited)
      return true
  return false
```

Write your solution in `getDfsOrder` in `answers.cpp`

Question 2: Alphabet Reconstruction (20 pts)

Linguistics researchers recently discovered a new language that uses an alphabet where each character is representable by a `char`, but the only remnants of the language is a part of a dictionary (the rest is possibly lost forever). The linguists enlisted your help to uncover the order of characters in the alphabet given what's left of the dictionary.

The dictionary is given to you as a list of words in the new language, meaning that the words are lexigraphically sorted *according*

to the order of the alphabet of that language. There is no concept of upper / lower case, i.e., you should treat the letter `a` and `A` as entirely different characters, and there may be characters other than `a-z`.

Example 1:

```
 we9r
wrrt
ewt
ett
etr
9ee
tew
```

From this word list, we deduce there are five (known) letters in the alphabet `w`, `e`, `9`, `r`, and `t`. Inspecting carefully, we see its order is `we9tr`

To devise a general purpose algorithm, think about how we deduced the characters in the alphabet and how we can deduce the order for the above example. The alphabet is simply all the characters that appear in the word list. As we process the word list, we want to build a "precedence graph," where each char in the alphabet is a node and there is a directed edge from X->Y if we know X is ahead of Y in the alphabet. Exactly how to tell X is ahead of Y from the word list is left for you to figure out! Once we have the precedence graph, we can run a topological sort to find out the ordering.

a. buildGraph (10 pt) Implement the function `Graph buildGraph(const vector<string>& wordList)`, which takes in the partial dictionary and builds the precedence graph where each char in the alphabet is a node and a directed edge X->Y means that X appears before Y in the alphabet. You may use the `Graph` methods `addEdge`, `hasEdge`, `addNode`, etc we've provided for you (be sure to check out their implementation too!).

Write your solution in `buildGraph` in `answers.cpp`.

b. topoSort (10 pt) Implement the topological sort function `string topoSort(const Graph& g)`, which returns a valid order of the entire alphabet in the form of a string. For Example 1, the return value should be the string `"we9tr"`.

You may assume that calls to this function will yield at most one alphabet ordering.

Write your solution in `topoSort` in `answers.cpp`.

Question 3 Island Hopping! (30 pt)

Josiah is thinking about becoming a cartographer! Hopping from Mykonos to Crete to Santorini and even to Ibiza (!!!), Josiah recorded where each island was relative to one another in Mediterranean Ocean. He looks to you and your COMP 285 algorithms knowledge for help in understanding his data.

a. (10 pt) The first map Josiah created was in the form of a `m x n` 2D binary grid. This terrain grid represents a map of true and false values denoting whether there is land or water present (respectively) at that particular location. He wants to write a program that can count the total number of distinct islands.

Josiah clarifies that a contiguous island can only be formed by cells horizontally or vertically adjacent; diagonally adjacent cells are not part of the same island.

For example, the following grid has 4 islands.

```
 terrain = [
[1, 0, 0, 0, 0],
[1, 0, 0, 1, 0],
[0, 0, 1, 1, 0],
[0, 1, 0, 0, 0],
[0, 0, 0, 0, 1]
]
```

The islands are the following groups of cells: 1. `terrain[0][0]`, `terrain[1][0]` 2. `terrain[1][3]`, `terrain[2][3]`, `terrain[2][2]` 3. `terrain[3][1]` 4. `terrain[4][4]`

Notice that `terrain[3][1]` and `terrain[2][2]` don't merge to form one island.

Write your solution in `numberOfIslands` in `answers.cpp`

b. (5 pt) Josiah wants to make sure the algorithm is efficient! He asks you, "What's the runtime of the `numberOfIslands` algorithm?" (in terms of m and/or n)

Write your answer in `q3.txt`

c. (10 pt) Now Josiah shows you an alternate map he created. Instead of a binary map of trues and falses, this map now contains values signifying the number of unique animal species he encountered during his exploration. He wants to know which island has the most biodiversity! More specifically, he wants to know the number of animal species you can find on that island! (He wants to know the island with the largest number of different animals. The program should return the total count of animal species on that island.)

For example, using the same example, your program should return 7.

```
terrain = [
[5, 0, 0, 0, 0],
[1, 0, 0, 4, 0],
[0, 0, 2, 1, 0],
[0, 3, 0, 0, 0],
[0, 0, 0, 0, 1]
]
```

The cells `terrain[1][3]`, `terrain[2][3]`, `terrain[2][2]` sum to 7 which is larger than the cumulative value of any other island.

He looks to you to come up with a solution!

Write your solution in `mostBiodiverseIsland` in `answers.cpp`

d. (5 pt) Again, Josiah wants to make sure the algorithm is efficient! He restates his question, "What's the runtime of the `mostBiodiverseIsland` algorithm?" (in terms of m and/or n)

Write your answer in `q3.txt`

Question 4: Real World Graph Applications (15 pts) For each of the following real-world problems, formulate each as a graph problem: - *What are the vertices?- What does an edge represent?- Do the edges have weights?- Is this undirected or directed?*

What algorithmic problem about graphs do we need to solve in order to solve the following problems? (Note, you don't actually have to solve these problems, just transform them into graph problems that we might then be able to solve).

a. (5 pt) Among actors, a 'Bacon number' is the number of degrees of separation from an actor to Kevin Bacon. For example, Kevin Bacon's Bacon number is `0`. If an actor works in a movie with Kevin Bacon, the actor's Bacon number is `1`. If an actor `A` works with an actor `B` who worked with Kevin Bacon in a movie, then actor `A`'s Bacon number is `2`, and so forth.

How would you represent this as a graph so we could answer questions like: - What is Samuel L. Jackson's Bacon number? - List all the people with Bacon number equal to 6.

Write your answer in `q4.txt`

b. (5 pt) You need to take a bunch of classes at North Carolina A&T University, and some of them depend on each other. For example, you must take `COMP 280` before taking `COMP 285`.

Given a set of classes you need to take, and information about which class is a pre-requisite for which other class, generate an order in which to take all of the classes. Assume you can only take one class at a time. How would you represent this as a graph so we can answer this question?

Write your answer in `q4.txt`

c. (5 pt) Believe it or not, you can make graph problems out of exchanging currencies! We want to detect whether or not we can make money out of thin air depending on the current exchange rates.

Given a set of currencies and their exchange rates to other currencies, how would we be able to tell if a cycle exists such that if you start with $1 of currency `C` and perform a series of exchanges such that you end up back at currency `C` with >$1. This would present a vulnerability we could exploit to make quick money!

Write your answer in `q4.txt`

Question 5: Technical Interview Practice, Career Prep Module #4 (20 pts)

The next step of our career preparation journey is getting ready for technical interviews! As we discussed in class, the only way to become skilled at technical interviews is to practice, practice, practice!

For this assignment, you will simulate a technical interview for yourself as you solve two coding questions.

1. Take at least 20 minutes per question and to pretend as if you were being asked to solve it in an interview setting.

- I recommend recording yourself so you can listen back and identify ways to improve for next time.

2. After your simulation, breakdown each problem down using the UMPIRE method as follows:

- **Understand:** Come up with 2 new test cases and what the program should return in each case.
- **Match:** List the data structures (if any) that you think will come in handy for your solution.
- **Plan / Pseudocode:** Write out your proposed approach in either pseudocode or paragraph form.
- **Implement:** Translate your pseudocode into actual code **without using Google for syntax, hints, etc.**.
- **Review:** If you came to a working solution, run through your program using both of the test cases you came up with earlier and note any bugs that you come across.
- **Evaluate:** Give the time and space complexity of your algorithm.

3. Copy and paste your code into the LeetCode IDE link provided, run it, and address any compiler errors until you have a working solution. Submit your code and take a screenshot of your accepted submission passing all test cases with the date and time shown.

- You'll upload this screenshot to your Repl.it HW5 directory so it's included in your final submission.

As an example, let's say we were simulating an interview where we were asked to **write a program that takes in a non-empty vector of integers and returns the average of the array.**

Here's an example response: - **Understand:** So if I was given `[4]`, my program should return `4` and if I was given `[1, 3]`, my program should return `1`. - **Match:** I will need an integer to keep track of the smallest element I've seen so far in the array. - **Plan / Pseudocode:** - Paragraph form: I'm going to initialize an integer called minSeenSoFar that will store the smallest value we've seen. Then I'll iterate over the vector and check to see if any of the elements are smaller than minSeenSoFar. If I find an element that is smaller, I'll update minSeenSoFar to be the new smallest value. At the end of the function, I'll return minSeenSoFar. - **Pseudocode form:**

```
def findSmallest(vec):
  min variable = INT_MAX
  for each element in vec
    if element is less than min
      update min variable
  return min variable
```

- **Implement:** Translate your pseudocode into actual C++ code without using Google for syntax, hints, etc..

```
int findSmallest(vector<int> vec) {
  int minSeenSoFar = INT_MAX
  for (int i = 0; i < vec.size; i+) {
    int element = vec[i];
    if (element < minSeenSoFar) {
      minSeenSoFar = element
    }
  }
  return element
}
```

(Note: if you look closely, the above code has a few syntax errors/bugs and that's okay! We're just practicing!) - **Review:** - **Test Case 1:** `vec = [4]`. We initialize `minSeenSoFar` to be INT_MAX then we check each element in the array. When `element = 4`, `element` is less than `minSeenSoFar` because `4 < INT_MAX` so we update `minSeenSoFar = 4`. Then at the end of the function we return `element` as our answer. - **Test Case 2:** `vec = [1, 3]`. We initialize minSeenSoFar to be INT_MAX then we check each element in the array. When `element = 1`, `1 < INT_MAX` so we update `minSeenSoFar = 1`. Then we look at the second element and see that `3` is not less than `minSeenSoFar` because `3 < 1` is `false` so we don't do anything. Then at the end of the function we return `element` as our answer. - **Evaluate:** The runtime is O(n) because we iterate over the vector and the space complexity is O(1).

At this point, I would have copied and pasted my solution into the designated LeetCode IDE and had to debug/resolve syntax errors. I would take a screenshot of my submitted solution and uploaded the photo to my HW5 Repl.it. The actual solution is here for your review:

```
int findSmallest(vector<int> vec) {
  int minSeenSoFar = INT_MAX;
  for (int i = 0; i < vec.size(); i++) {
    int element = vec[i];
    if (element < minSeenSoFar) {
      minSeenSoFar = element;
    }
  }
  return minSeenSoFar
}
```

a. (10 pt) LENGTH OF LAST WORD

Given a string `s` consisting of words and spaces, return the length of the last word in the string. For example:

```
when s = "hi there", the program should return 5 because the last word is "there" with length 5.
when s = "   what are   you   doing here  ", the algorithm should return 4 because the last word is "h
```

Write your answers in `q5.txt`

Copy and paste your written code into here and modify as necessary until you have working solution. Upload screenshot of your accepted submission and name it 5a.png (make sure it's a png)

b. (10 pt) PLUS ONE

You are given an integer represented as an integer array `digits`, where each `digits[i]` is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.

Increment the integer by one and return the resulting array of digits.

For example:

```
when digits = [1,2,3], the program should return [1,2,4] because 123 + 1 = 124
when digits = [9], the program should return [1, 0] because 9 + 1 = 10
```

Write your answers in `q5.txt`

Copy and paste your written code into here and modify as necessary until you have working solution. Upload screenshot of your accepted submission and name it 5b.png (make sure it's a png)