CSCI 3104 PS5b

Luna Mcbride

TOTAL POINTS

30 / 48

QUESTION 1

25 pts

1.1 3 / 5

+ **2 pts** Iterating over all edges and merging them

✓ **+ 3 pts** **Checking for cycle formation while adding edges**

+ **0 pts** Not attempted

+ **0 pts** Incorrect

+ **0 pts** (Non-scored rubric item) You have used constants like 127 in place of no_of_vertices in your code. Use len(vertices).

+ **0 pts** Stopping the iteration once you add n-k-1 edges to the tree. This does not guarantee to find correct MST when k>0. The logic for finding MST is not completely correct.

- **2 pts** The submitted code gave error while trying to execute it. Had to make some changes to run the code. Check your code before submission

- **5 pts** Results in pdf are not supported by code

+ **0 pts** The submitted could not be executed as it was throwing several errors.

Stops merging edges with components when number of components = max value of k. This results in an incomplete application of Kruskal's algorithm and an incorrect answer

1.2 10 / 10

✓ **+ 5 pts** **Updating the leader of the nodes in one component with another component's leader.**

✓ **+ 5 pts** **Adding the nodes in one component to another component. (Adding nodes to larger component or another component)**

- **2 pts** Using a smaller component's leader to update the leader of nodes in the large component, instead of doing reverse. This does not guarantee O(mlogn) time.

- **2 pts** Joined larger components to smaller ones instead of joining smaller components to larger ones to gain extra optimization.

+ **0 pts** (Non-scored rubric item) Deleting the nodes in smaller components. Deleting them from memory leads to efficient implementation.

+ **0 pts** Not attempted

+ **0 pts** Incorrect answer

+ **2 pts** You are not updating the leaders of nodes in smaller component. This is leading to incorrect result.

+ **4 pts** In your solution, you are not updating leaders of nodes in the smaller component and you are not merging all nodes in the smaller component with larger component. The implementation is incorrect.

- **1 pts** The code was not executable but could be executed with a small modification.

- **3 pts** The submitted code gave error while trying to execute it. Had to make some changes to run the code. Check your code before submission

+ **0 pts** The submitted could not be executed as it was throwing several errors.

1.3 0 / 10

+ **6 pts** Correct code for finding clustering spacing

+ **4 pts** Correct results written (Printing the correct spacing values for all k)

+ **2 pts** Result (printing of spacing values) does not have correct spacing values for all k.

- **1 pts** Not printing the spacing values in the correct format

✓ **+ 0 pts** **Incorrect or incomplete answer**

+ **0 pts** Not attempted

- **1 pts** Expected to run for all k between 2...10 and

not just one value of k in the code itself.

+ **0 pts** Your code is finding the spacing values for k-1, not for k

+ **3 pts** Logic was implemented in the exact reverse way but code to find spacing values was placed correctly.

✓ **- 10 pts** **Results without supporting code**

+ **0 pts** You are stopping the iteration (or not adding further edges) once you add n-k-1 edges to the tree. This does not guarantee to find correct MST when k>0. The logic for finding MST is not completely correct.

+ **0 pts** (Non-scored rubric item) You can compute all spacing values for all k while you compute MST using Kruskal's algorithm. No need to do it separately. See the solution.

+ **4 pts** Written code for finding the spacing value of k+1 (or k-1) instead of k. This is leading to incorrect result

- **3 pts** Your code was throwing errors when executed. Had to make some changes for executing it. Make sure to submit a working code.

+ **0 pts** The submitted could not be executed as it was throwing several errors.

Unclear how values for k are printed when code does not generate it.

QUESTION 2

2 3 / 3

✓ **+ 1.5 pts** **Correct reasoning**

✓ **+ 1.5 pts** **Correct number of components.**

+ **1 pts** Incorrect number of components, but off by only a small quantity

+ **0 pts** No answer

+ **1 pts** There are n vertices in the graph, not k vertices. So there are k+1 components, not n+1 components

+ **0.5 pts** Recognized that Kruskal's algorithm combines components

+ **0 pts** Incorrect answer with incorrect or no reasoning

QUESTION 3

3 0 / 5

+ **5 pts** Correct

+ **0 pts** No or insufficient answer

+ **4.5 pts** Minor errors with weights

✓ **+ 0 pts** **Incorrect answer**

+ **2.5 pts** Correct codes, but did not draw tree

+ **0 pts** Plagiarism

+ **3 pts** Adjacency relation not clear.

We should have a node connected 3 and v_2. See solution.

QUESTION 4

4 5 / 5

✓ **+ 5 pts** **Correct (exactly pinpointed that S is a prefix of i or c is a prefix of e)**

+ **0 pts** Totally Incorrect

+ **3 pts** Kind of got the idea right but did not pinpoint the exact answer. (Need to clearly specify why the tree cannot be constructed)

QUESTION 5

10 pts

5.1 3 / 3

✓ **+ 1 pts** **Use the reasonable data structure.**

✓ **+ 2 pts** **Described correctly what's the vertex and how the vertex will be connected.**

+ **0 pts** Incorrect/Not attempted.

+ **2 pts** Union Find data structure used correctly.

5.2 2 / 2

✓ **+ 1 pts** **Traverse the graph/matrix throughly.**

✓ **+ 1 pts** **Define 0 as the vertex correctly. And connect vertex(i, j) correctly with other vertices.**

+ **0 pts** Incorrect/Not attempted.

5.3 4 / 5

✓ **+ 1 pts** **Explore the matrix/graph throughly.**

✓ **+ 2 pts** **Connect the vertex with other vertices correctly. (Consider the diagonal case)**

+ **1 pts** Consider the edge case correctly. (dealing

with i, j gently.)

✓ **+ 1 pts** **Store the results correctly.**

+ 0 pts Incorrect/Not attempted. (If you use union-find, you can see your grade and comments in the point adjustment.)

- 1 pts DFS/BFS should be done recursively to explore all connected 0s.

- 2 pts The algorithm was supposed to be solved for a general case and not a specific case.

The detail you provided is not enough. EX: how to define adjacent?

Name: Luna McBride
ID: 107607144

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 5b (48 points) **Fall 2019, CU-Boulder**

Instructions for submitting your solution:

- The solutions **should be typed** and we cannot accept hand-written solutions. Here's a short intro to Latex.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Gradescope if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.
- Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.
- For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.
- You may work with other students. However, **all solutions must be written independently and in your own words.** Referencing solutions of any sort is strictly prohibited. You must explicitly cite any sources, as well as any collaborators.

Name: Luna McBride
ID: 107607144

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 5b (48 points) **Fall 2019, CU-Boulder**

1. (25 pts) For this question, you are going to implement Kruskal's algorithm and union-find to build an MST from supplied data. Refer to the python starter code **MST_Q1_starter_code.py** on Canvas that generates a graph of US cities, where the cities are the vertices and the edges are the distances between them. The code requires **miles_dat.txt.gz** file as the graph data source so keep it in the same folder as the code. Before you start writing any code, make sure you can build the code that's been supplied. The code uses the networkx library. You may need to install this library for the code to run.

   Read all instructions for this question carefully.

   (a) (5 pts) Complete the code to find the edges that are part of the MST. You should add these edges in the list *kruskal_selected_edges*. Do not change the existing format of the edges. They are represented as a tuple of vertices and a vertex is represented like v = "Waukegan, IL". Read the comments in the code for more information. You don't need to read/understand the *miles_graph*() and *draw_graph*() functions.

   (b) (10 pts) Implement the *union*() function to implement Kruskal's.

   (c) (10 pts) Modify your code slightly so that you can produce disconnected components. Let's call these components clusters. The "spacing" of any particular clustering (group of clusters) is defined as the smallest edge between vertices in any pair of different clusters. If we stop Kruskal's k iterations before the algorithm completes, what is the spacing value? Run your code for $k = 2...10$ to generate spacing for all these k values. Your code needs to have this calculation for your answer to receive credit.

   (d) In the pdf that you submit for this assignment, please include the following:

      i. One of the generated graphs **MST.png** that your code produces that shows the MST for that run. Note that on each run, you can get a different number of edges to begin with. Thus, you can expect a different answer each time you run.

      ii. The spacing values for each k value that you use.

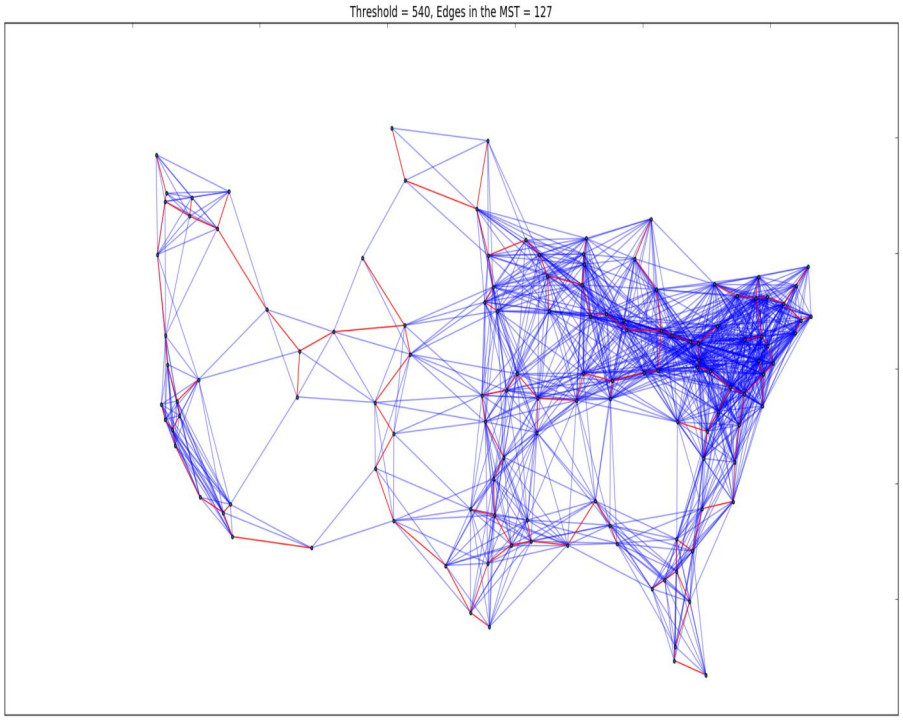
      iii. Your .py file for this question needs to be submitted to Canvas.

Name: Luna McBride
ID: 107607144

(Space for Q1 image and spacing values)

Solution.

Spacing value for each k:
[('Santa Barbara, CA', 'Salinas, CA') ('Sterling, CO', 'Salida, CO'), ('Valley City, ND', 'Winnipeg, MB'), ('Roswell, NM', 'San Angelo, TX'), ('Salem, OR', 'Weed, CA'), ('Sheridan, WY', 'Scottsbluff, NE'), ('Williston, ND', 'Valley City, ND'), ('Scottsbluff, NE', 'Rock Springs, WY'), ('Twin Falls, ID', 'Walla Walla, WA'), ('San Diego, CA', 'Tucson, AZ')]



Name: Luna McBride
ID: 107607144

2. (3 pts) How many disconnected components are there when you stop Kruskal's k round before you complete the MST? Justify your answer.

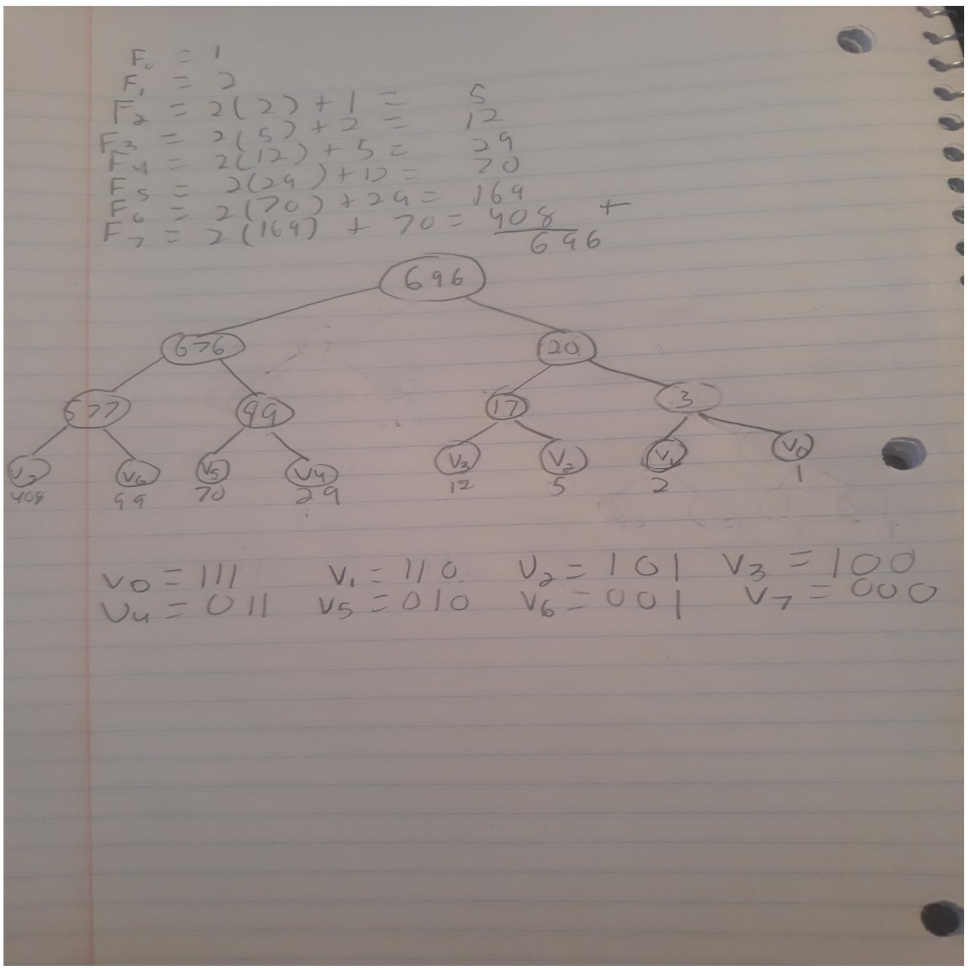
   Solution.
   For k iterations before Kruskal's stops, there are k+1 discomponents left. Since union connects 2 disconnected graphs, each iteration takes one graph out of the equation. That means for every run of the algorithm, one disconnected component is removed. We are looking for one connected whole at k=0 (the end of kruskals), and since the union brings 2 into 1, the step before must have 2 graphs. k=0− >components=1, k=1− >components=2. Therefore, k=k− >components=k+1

3. (5 pts) Consider the recurrence $F_n = 2F_{n-1} + F_{n-2}$, with the base cases $F_0 = 1$ and $F_1 = 2$. Suppose we have letters $v_0, \ldots, v_7$; where for $i \in \{0, \ldots, 7\}$, the frequency of v_i is given by F_i. Draw a Huffman tree for $v_0, \ldots, v_7$.

   Solution.

Name: Luna McBride
ID: 107607144

CSCI 3104, Algorithms **Profs. Hoenigman & Agrawal**
Problem Set 5b (48 points) **Fall 2019, CU-Boulder**



4. (5 pts) Assume you run your Huffman tree algorithm and you produce the following pre-fix codes. Describe why there must be an error in your algorithm.

```
S = 00
c = 01
i = 001
e = 011
n = 101
```

Solution.
Let us take the value 001. This looks like it is 'i', but for all we know, it could be 'S' with the beginning part of 'n'. Imagine this in a very long string where these sort of issues persist.
Since Huffman trees are meant to prevent this ambiguity, there had to have been an error in the algorithm

Name: Luna McBride
ID: 107607144

5. (10 pts) Assume you're given an integer matrix that represents a plot of land, where the value at that location in the matrix represents the height above sea level. A value of zero indicates water. A pond is a region of water connected vertically, horizontally, or diagonally. The size of the pond is the total number of connected water cells. Write an algorithm to compute the sizes of all ponds in the matrix.

   Example:

   ```
   0 2 1 0
   0 1 0 1
   1 1 0 1
   0 1 0 1
   ```

   would output 1, 2, 4.

   (a) (3 pts) Describe the graph data structure that your algorithm will use for this problem.

   Solution.
   For this, I will only be counting the 0's in our graph. This is an unweighted, undirected graph connecting adjacent 0's. Each pond would be a disconnected component in terms of the graph, just as the map works in question 1 (The area not considered on the US map exists, but it is not considered in that circumstance), for example.

(b) (2 pts) Provide a 3-4 sentence description of how your algorithm works, including how the matrix is converted to the graph, how adjacent vertices are identified, and how the algorithm traverses the graph to identify connected vertices.

Solution.
The algorithm will take the array, read through it, and put every 0 as its own component. It will be holding the place in the array it exists in as well. The algorithm will then go through the list of 0's we have disconnected and look for those that are around it (for [1,1], the adjacent are all possibilities of adding or subtracting 1 from the 2d array, so [0,0],[0,1],[0,2],[1,0],[1,2],[2,0],[2,1],[2,2]). It will then go through to find the amount of ponds, then return them and their sizes via a list.

(c) (5 pts) Write an algorithm to solve this problem.

Solution.
pond(arr):
−− > for i in range(0,len(width of array))
−−/−− > for j in range(0,len(width of array))
−−/−−/−− > if [i,j] is a 0
−−/−−/−−/−− > make it its own node, store the [i,j] value, store the 0 in a list of 0's

−− > while list of 0's is not empty
−−/−− > check adjacent array values for 0
−−/−− > if there is a 0
−−/−−/−− > connect the nodes, make new adjacencies for them as a group, reinsert group to be checked for surrounding 0's

−−/−− > else
−−/−−/−− > remove the 0's from the 0's list, add the size of the group to the solution list

−− > return solution list