Homework 4 – Graphs

Arya Rahmanian

COMP 614

November 1, 2024

1. CodeSkulptor Link: https://py3.codeskulptor.org/#user309_vYuo8EaD7g_0.py

2.

a. **Question 3.D.i: The weakly connected components definition requires that your algorithm treat a directed graph as an undirected graph. How will you accomplish this? Be as specific as possible!**

To treat the directed graph as an undirected graph, I will implement a helper function called graph_as_undirected(graph). This function will create a new graph that represents the undirected version of the original directed graph without mutating the input graph. I will start by making a deep copy of the original directed graph using the class function, .copy(). Then, I will add reverse edges to make the graph undirected. I will do this by iterating through each neighbor to a node and will add an edge back from the neighbor to the node.

b. **Question 3.D.ii: How can you use BFS to help you to solve this problem? You should provide a detailed description of how you will choose which node(s) you will use as the start node(s) when you run BFS, as well as how you will use the results of your call(s) to BFS to get the connected components. To improve the clarity of your answer, I would strongly recommend that you refer to specific data structures that you intend to create and use within your algorithm.**

I will use BFS to identify the nodes that are part of the same weakly connected component by exploring the undirected graph from a start node. I will start BFS from unvisited nodes to find new weakly connected components. I will use two python sets: visited and components. Visited will track nodes I have processed, and components will store the connected nodes. Lastly, I will also use a dictionary from the BFS to identify the reachable nodes. Here is the overall algorithm process:

- Iterate through all nodes.
- For each unvisited node, perform BFS on the undirected graph.
- Collected all reachable nodes as a connected component.
- Update visited and components.
- Repeat until all nodes are visited.

3.

a. **Run BFS on a DiGraph built from wikipedia_articles.txt, and find the distance from "rice university" to "university of florida" as well as the distance from "university of florida" to "rice university". What were the results? What chains of nodes were traversed in order to get from "rice university" to "university of florida" and vice versa? Do the results that you got make sense? Why or why not?**

Distance from Rice University to University of Florida: 3

Path from Rice University to University of Florida: rice university -> super bowl viii -> new york jets -> university of florida

Distance from University of Florida to Rice University: 2

Path from University of Florida to Rice University: university of florida -> university of virginia -> rice university

The results do make sense since Wikipedia articles link are directed and the path between two different topics may have different paths going in both directions. Going from UofF to Rice makes sense as they connected through University of Virginia. At first glance, “rice university -> super bowl viii” didn’t make sense until I realized that super bowl was at Rice Stadium, pretty cool!

b. **Compute the weakly connected components for the DiGraph built from wikipedia_articles.txt. How many connected components are there? Are these the results that you expected? Why or why not? Why do you think you got these results? How do you think things would have differed if you had computed the strongly connected components instead?**

Number of weakly connected components: 1

Although this is not exactly what I expected, it does make sense since Wikipedia articles incorporate as many links as possible onto its pages to try to interlink them as much as possible. So, this means with all the articles we had in our graph, they are all connected somehow if we traverse the graph enough. This shows how much the platform tries to cross reference different topics.

On the other hand, if we searched for strongly connected components, we would get a very different number. This would require nodes to have bidirectional connectivity, which we did not see even between Rice University and University of Florida. The strongly connected components number for this graph would at least be in the hundreds for this graph.

c. **Running a graph search algorithm and finding connected components are two possible approaches to gaining insight into the similarity between articles, but they're not the only possible approaches. What could be another viable approach to analyzing the similarity between articles that doesn't involve a graph search algorithm or finding connected components? You do not need to go into great detail; a high-level description is sufficient. There is not a single right answer here! Just make sure to justify why your approach is reasonable.**

We could develop some sort of k means clustering algorithm to group articles together based on their word embeddings used in NLP. Using a pretrained embeddings table, we can convert the articles into vector tables and then use these tables to group together different articles.

Reflection

1. **What were the two most important concepts and/or skills that were reinforced by this assignment? Why do you believe that these concepts are important?**

   The most important skill reinforced in this assignment is BFS. BFS is widely used in graph and tree search and analysis. It is an essential algorithm that most developers and data scientists should be aware of and use when necessary. Another important concept used in this assignment is graphs. Graphs are an important data structure used in computer science to help model relationships between data points and help programmers understand their data as well.

2. **How do the skills and concepts that you applied in this assignment transcend the specific application/problem that you were tackling? How could you envision applying these skills and concepts again in the future?**

   Like I mentioned above, both Graphs and BFS are hand in hand in fundamental data structures and algorithms that computer and data scientists need to understand. Graphs are very useful in understanding relationships between data points and BFS is an efficient and widely used algorithm to search these graphs or trees.

3. **What do you believe you did well on this assignment? If you could do this assignment over, would you do anything differently? Why or why not?**

   I believe I did well in understanding BFS and graphs well enough to develop a good algorithm to find all weakly connected components in the graph. Although it was hard at first, I was able to grasp a good understanding of the problem and come up with a good algorithm to find all connected components. If I did the assignment over, I wouldn't change anything, since I am happy with my result.

4. **Do you think you're comfortable enough with the concepts covered in this assignment that you would be capable of teaching them to a peer? Why or why not?**

   I am comfortable with the concepts covered in this assignment, graphs and bfs. But I don’t think I understand them well enough to teach them to a peer. Although I felt successful in this context, it took me a while to really understand the problem well enough to develop a solution.