CS 2302

Fall 2019

Lab Report #7

John Rodriguez

Due: December 6, 2019

Professor: Olac Fuentes

TA: Anindita Nath

Introduction

For lab 7 we were given the task of first implementing the Randomized Hamiltonian Function which is used to determine if a graph contains a Hamiltonian Cycle. We were given the pseudocode for this function, but we had to interpret it into python. The second part of the lab was to again create a Hamiltonian Cycle function, but this time implement it using backtracking instead of randomization. Finally, we must modify the Edit Distance function given to us to allow replacements only in the case where the characters being interchanged are both vowels, or both consonants.

Proposed Solution Design and Implementation

I approached this lab by first focusing on implementing The Randomized Hamiltonian Cycle function. To implement it into my program I used the pseudocode given to us as a reference. The function has the parameters of the graph and the amount of times you want to use randomization. I then converted the graph to an edge list and then created an adjacency list and inserted random edges from the edge list into the adjacency list and used the connected components function to see if it’s a Hamiltonian cycle. I then used an in-degree function to make sure the indegree of the vertices is 2.

I then worked of the Backtracking Hamiltonian cycle part of the lab. For this function I again converted the graph to an edge list and the only parameter is a graph but created a helper function. I first checked if the number of edges matched the number of vertices and then convert the edge list into an adjacency list and then check if the in-degree for each of the vertices is 2 and then it returns if the graph is a Hamiltonian Cycle or not.

The last part of the lab I worked on was changing the edit distance function given to us to allow replacements only in the case where the characters being interchanged are both vowels, or both consonants. To do this I created a variable called vowel which was used to check if both letters from both words were vowels. I then checked if both the letter were not vowels if both the letters were consonants and used the vowel variable, I created to also check this. If one these conditions is met, then the function will behave as the normal edit distance function.

I also created two test functions for each Hamiltonian Cycle functions. I also created a menu in which you can choose to test the Randomized Hamiltonian Cycle, Backtracking Hamiltonian Cycle, or the modified edit distance function.

Experimental Results

**Test for Randomized Hamiltonian Cycle:**

**A close up of text on a white background

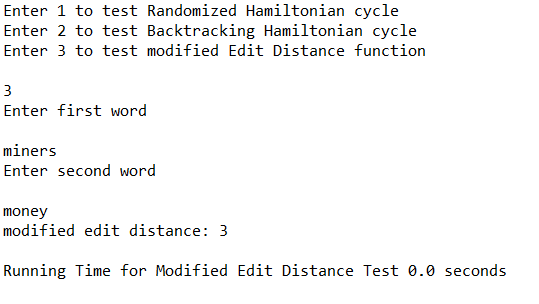
Description automatically generated**

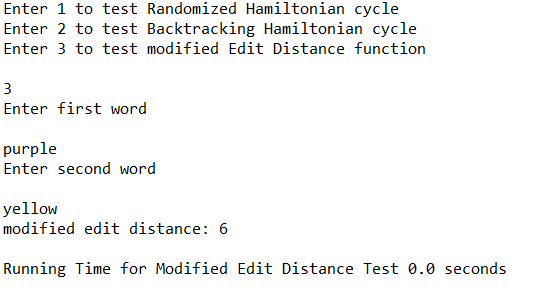
**Test for Backtracking Hamiltonian Cycle:**

**A close up of text on a white background

Description automatically generated**

**Tests for Modified Edit Distance:**





**Running Times:**

|  |  |
| --- | --- |
| FUNCTION TEST | RUNNING TIME |
| Randomized Hamiltonian Cycle | 0.015622615814208984 seconds |
| Backtracking Hamiltonian Cycle | 0.015619754791259766 seconds |
| 1st Modified Edit Distance | 0.0 seconds |
| 2nd Modified Edit Distance | 0.0 seconds |

Conclusion

This lab helped me become more comfortable implementing algorithm design techniques specifically the three used in this lab being randomization, backtracking, and dynamic programing. I also more about the two NP-complete problems used in this lab which were a Hamiltonian Cycle and Edit Distance. In this lab we were also given pseudocode for the Randomized Hamiltonian Cycle, so I had to interpret it into my python program. I feel like I’m a much better programmer after this lab and all the labs this semester.

Appendix

**graph\_AL.py**

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**graph\_EL.py**

**A screenshot of a social media post

Description automatically generated**

**A screenshot of a social media post

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**dsf.py**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a social media post

Description automatically generated**

**Lab 7.py**

**A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated**

**A screenshot of a social media post

Description automatically generated**

**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class