CS 2302 - Lab 6

Sofia Gutierrez

Olac Fuentes

**Introduction**

The objective of this lab was to create a program that would build and modify graphs using an adjacency list, adjacency matrix, and an edge list. This program would be composed of these three types of graphs which would then contain the appropriate functions needed to insert the necessary attributes (vertices, edges, weight). Along with the function to insert came other similar functions, such as deletion and display, and finally it would contain Breadth-first-search and depth-first-search. Our next and final objective was to create a graph using AL, AM, and EL, in order to display the final outcome of a riddle that was to be solved. The correct printed path- which would be the solution to the riddle and would be solved with the help of BFS and DFS- would then be printed for the user.

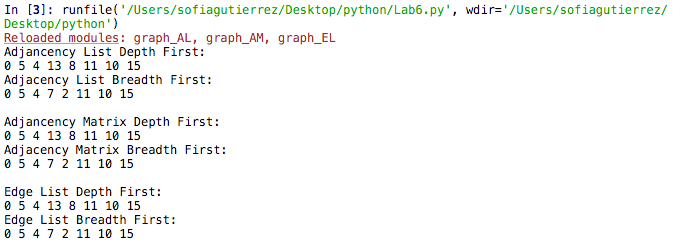
**Design and Implementation**

The task was to first complete the implementations of insert for every edge, delete for every edge, and to display the graph for edge list and adjacency matrix. Some of these materials were already premade so the task of completing these classes became easier. Some other functions included in these classes were to allow the user to go from one graph constructed to another. For example, use the graph built on the AL file but to transform it to EL. This was for the purpose of efficiency. Then came the functions Breadth-first-search and Depth-first-search which was easy to complete considering the pseudocode was provided to the author through Zybooks.

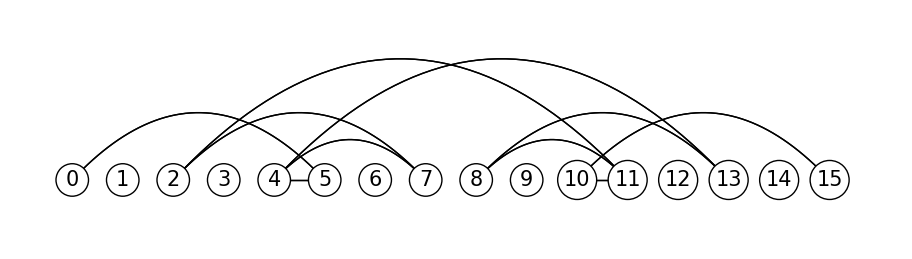
For the second part of the assignment the author was given a riddle. This riddle had simple rules in order to achieve what was needed to be achieved which was basically to create a path of numbers, focusing on their binary forms. The path started with 0 and from there, their binary digits were taken into consideration so that the number that came after 0 would follow the rules.

**Experimental Results**

The following is what resulted in when choosing the correct path. In this case it shows the path from start at vertex 0, which has a binary digit of 0000, and ending at 15, which has a binary digit of 1111.



The path was then drawn:



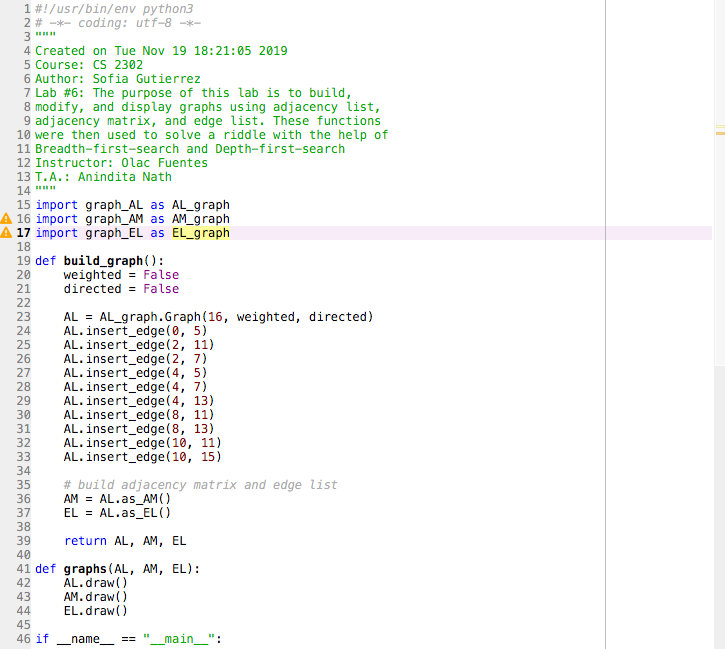
Big O Notation:

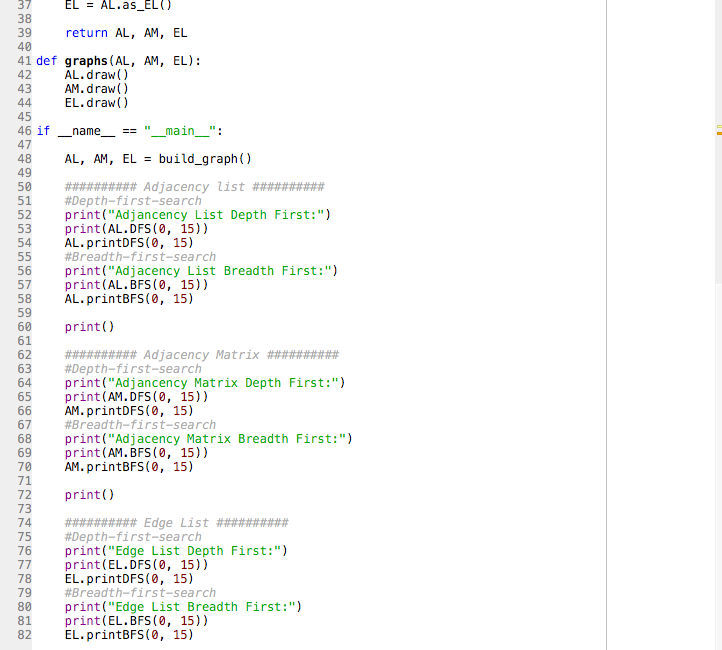
|  |  |
| --- | --- |
| Breadth-first-search | Depth-first-search |
| O(|v| + |e|) | O(|v| + |e|) |

**Conclusion**

Graphs can be displayed in a total of three ways, adjacency list, adjacency matrix, and edge list. I became more familiar which these three types of implementations and I also became more comfortable with BFS and DFS. The functions that were especially difficult to complete were the functions to transform one graph using one implementation to a graph using a different implementation.

**Appendix**





**Academic Honesty Certification**

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.

Sofia Gutierrez November 25, 2019