A5- GRAPHS

CSC2001F

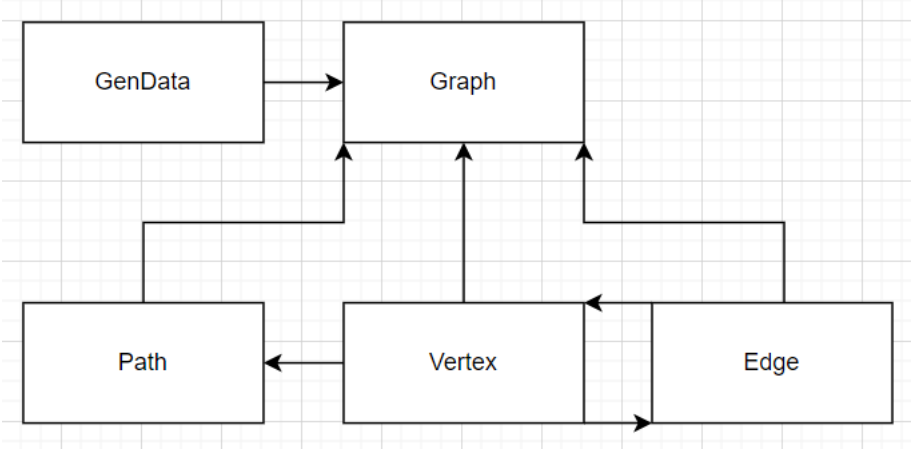
Zuhayr Loonat
LNTZUH001

Description

This assignment programmatically compares the performance of Dijkstra's shortest paths algorithm with the theoretical performance bounds.

This application generates data, inserts the data into a graph and measures the performance experimentally. It cycles through different combinations of number of Vertices and Edges.

Classes



Classes from least to most dependant.

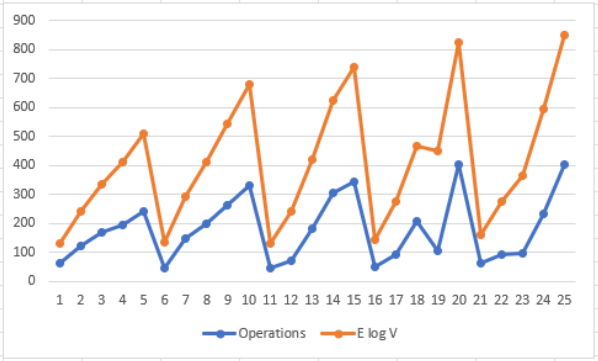
- GenData
- Edge
- Vertex
- Path
- Graph

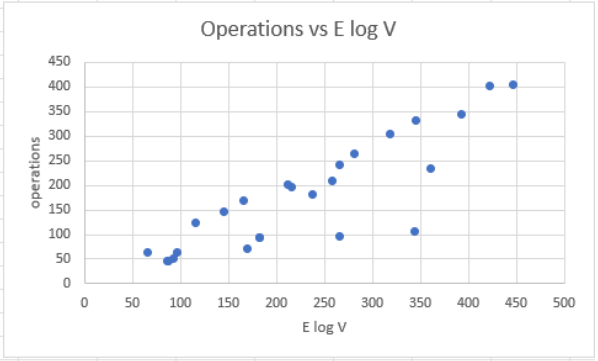
GenData: Class that creates dataset
Edge: Class for graph edges
Vertex: Class for graph vertices
Path: Class for the paths in the graph
Graph: Main class that that run Dijkstras Algorithm and outputs to a csv file.

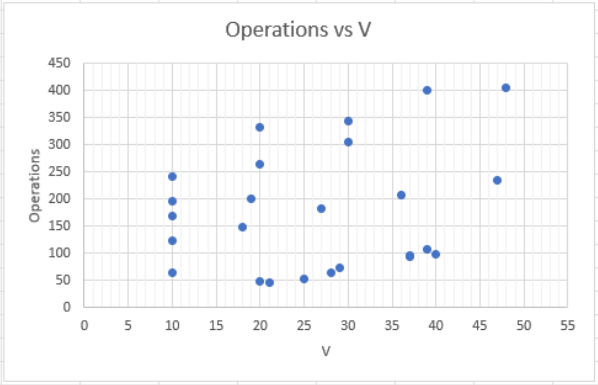
Results

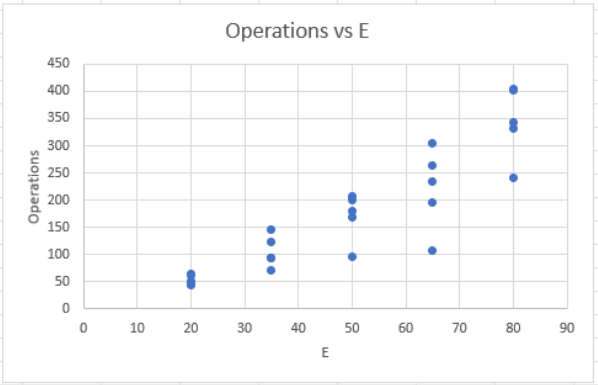
Shown below is the results of running the program and calculations done thereafter.

V	E	cV	cE	cPQ	Operations	E log V
10	20	8	15	12	65	66.43856
10	35	10	35	34	124	116.2675
10	50	10	50	48	168	166.0964
10	65	10	65	46	196	215.9253
10	80	10	80	62	242	265.7542
20	20	3	2	2	47	86.43856
18	35	16	32	46	147	145.9474
19	50	18	44	70	201	212.3964
20	65	20	65	94	264	280.9253
20	80	20	80	132	332	345.7542
21	20	2	2	0	45	87.84635
29	35	4	4	0	72	170.0293
27	50	17	27	60	181	237.7444
30	65	28	62	120	305	318.9479
30	80	29	77	128	344	392.5512
25	20	3	2	2	52	92.87712
37	35	6	5	10	93	182.3309
36	50	21	29	72	208	258.4963
39	65	2	1	0	107	343.5511
39	80	35	71	176	401	422.8322
28	20	6	5	4	63	96.1471
37	35	9	8	6	95	182.3309
40	50	4	3	0	97	266.0964
47	65	24	31	68	235	361.0483
48	80	38	62	176	404	446.797









Discussion of Results

From the picture above, by looking at the graphs we can come to find out a lot.

As we increase the umber of vertices the number of operations increase as well, but not as much as when you increase the number of edges. The number of edges has a greater impact on the number of operations needed since it means there are more paths.

Creativity

For the creativity in my application, I decided that instead of making multiple different files container data and then having to store them and the read them all separately, I created my program so that when you run the main file Graphs, it will create an output csv file for all 25 scenarios but will only use 1 data.txt file. I did this by running a function that generates the data through a nested for loop and then reading it in for each scenario, appending the results to the csv, and then moving on to the next scenario, and doing it again…
This saves space and is better than having your program open multiple files, and will allow your program to have to store less files in memory, making it faster.

Git log

```
$ git log | (ln=0; while read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
0: commit b1e236bb8c7b0090a278f0ca61f8a1343a97304b
1: Author: zuhayr.py <zuhayr.loonat@gmail.com>
2: Date: Sat May 6 21:02:04 2023 +0200
3:
4: renamed folder
5:
6: commit d10dfcb3b84f09cfa56d8f53dc40c6f65a0147e1
7: Author: zuhayr.py <zuhayr.loonat@gmail.com>
8: Date: Sat May 6 21:01:22 2023 +0200
9:
...
61: Author: zuhayr.py <zuhayr.loonat@gmail.com>
62: Date: Fri May 5 11:27:42 2023 +0200
63:
64: First commit
65:
66: commit 9f0ffae69494c72b496208868656988e3ae7900c
67: Author: zuhayr.py <119530173+zuhayrl@users.noreply.github.com>
68: Date: Fri May 5 10:06:57 2023 +0200
69:
70: Initial commit
```

https://github.com/zuhayrl/Graphs_CSC