Matthew Schramm Assignment 5 report

1. **What my OO design is:**

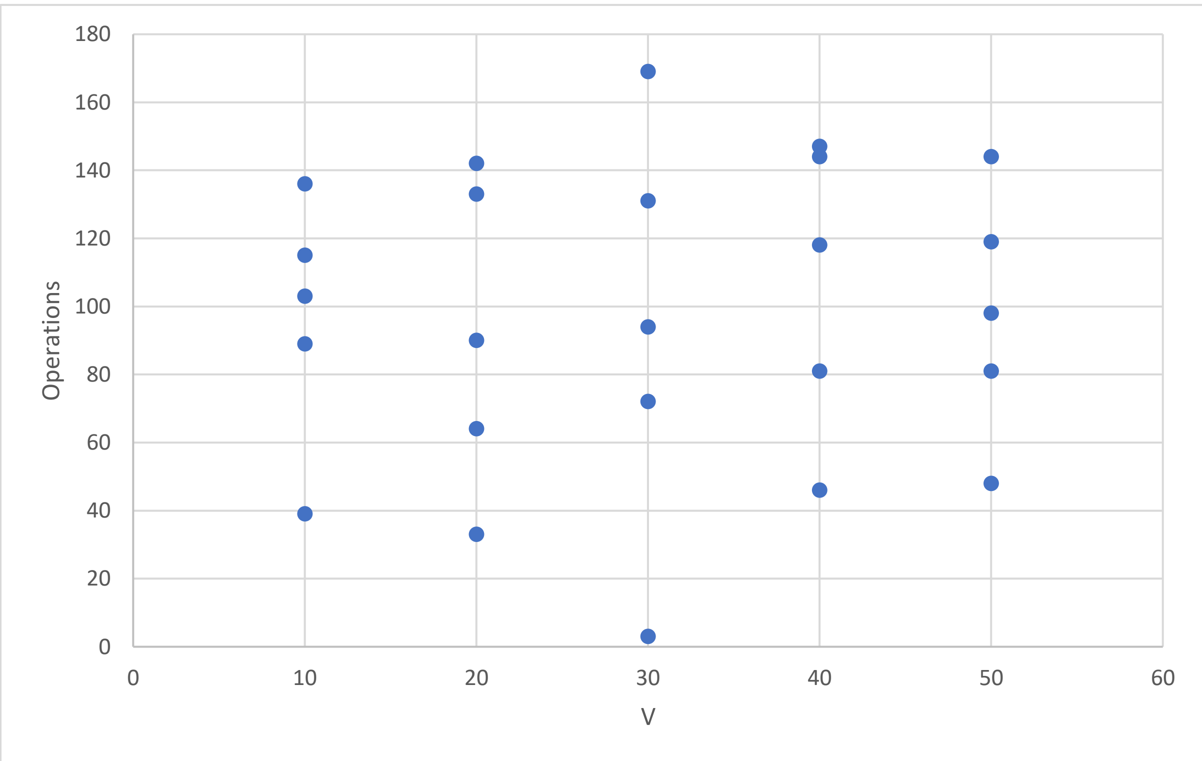
   My OO design consists of 6 classes all interacting with each other in a sequential way to create a functioning graph that we will be using to perform Dijkstra's method on a generated dataset. The edge and vertex class are the base classes of the design with them creating the various vertex and edge objects that will be inside the graph. Following these 2 classes is the path class which uses both the vertex and edge classes to create and output a path that is taken through the graph created and then finally the graph class which uses the Vertex, edge and path class functions and objects to create a working Graph that can perform, Dijkstra's algorithm, print the path that is taken and overall allow for data to be abstracted for use in calculations such as the one we are doing in this assignment lastly we have the GraphException class which allows for various exceptions to be output and fixed that may occur in the graph class. Then we have the GraphExperiment class that contains the main method of the design, which I created to generate the datasets that will be used in the various calculations, the class once generated the dataset will add to the various nodes containing vertices and edges to a new graph object where Dijkstra's algorithm is performed allowing for the complexity to be calculated. The entirety of the problem that we are addressing in this OO design is solved through the running of the GraphExperiment class which through encapsulation uses all the classes mentioned previously.
2. **What the goal of the experiment is and how you executed the experiment. Report on any experiment design decisions you made.**
   The goal of the experiment is to see how well the Dijkstra algorithm conform to our expectation in terms of its performance or in other words we have this theoretical bounds of performance and we want to see through generation of datasets and running of the algorithm does this fit this theoretical bound of performance. We do this through the creation of a class that will generate datasets based on a certain vertices value and edges value which will then be inputted into the graph class where a graph will be created. Once we have a graph object containing the nodes of the generated dataset we perform Dijkstra's algorithm on it where we have instrumentation that counts the number of comparisons that are performed throughout the algorithm increment a vertices count variable. This data is then stored and output to a CSV file where graphs are created showing the bounds of the complexity E log V in action and allowing us to deduct and prove that the algorithm does fit this theoretical bound of performance.

3. **What the goal of the experiment is and how you executed the experiment. Report on any experiment design decisions you made.**
    - <u>Results of very small values of edges with instrumentation that counts the processing of vertices, edges and the priority.</u>

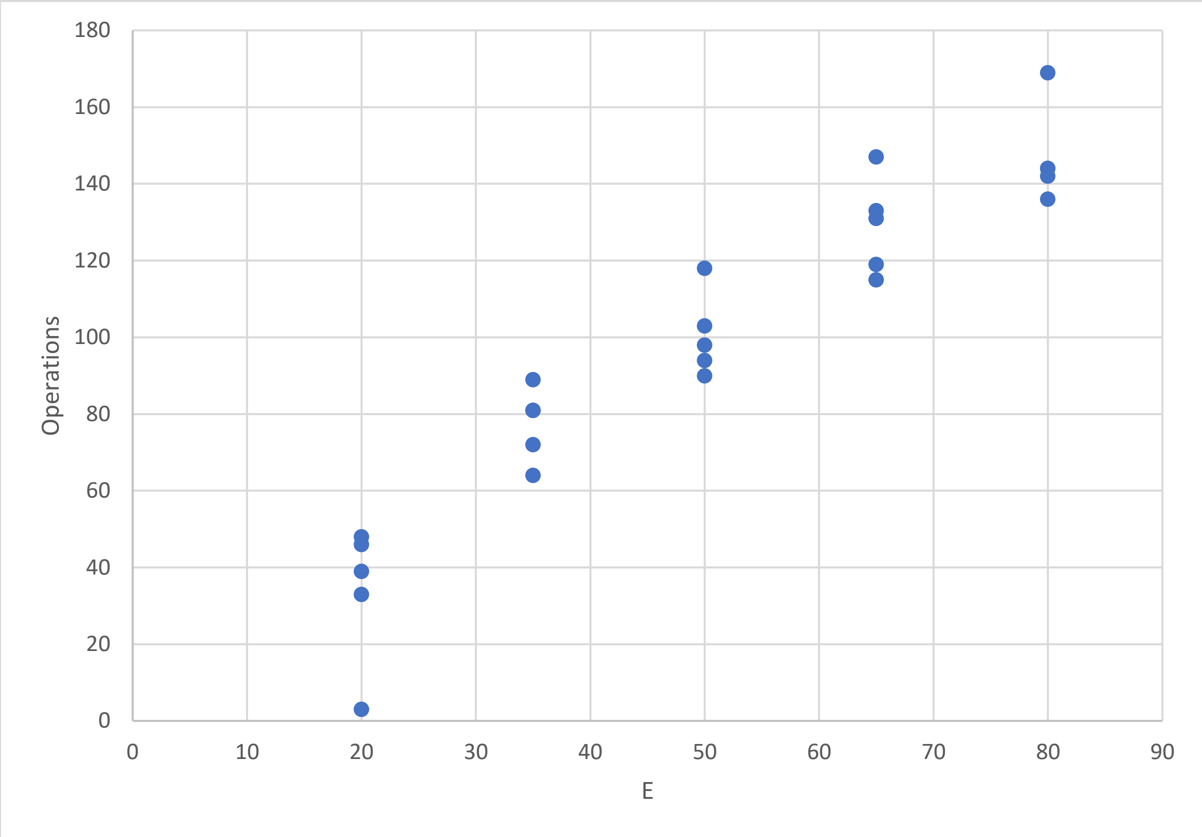
Table 1: To show the values of the data that is being used in the experiment

V	E	Vc	Ec	Pc	ElogV	Operations
10	20	8	15	16	66,4385619	39
10	35	10	35	44	116,2674833	89
10	50	10	50	43	166,0964047	103
10	65	10	65	40	215,9253262	115
10	80	10	80	46	265,7542476	136
20	20	7	16	10	86,4385619	33
20	35	9	33	22	151,2674833	64
20	50	10	50	30	216,0964047	90
20	65	10	65	58	280,9253262	133
20	80	10	80	52	345,7542476	142
30	20	2	1	0	98,13781191	3
30	35	9	31	32	171,7411708	72
30	50	10	50	34	245,3445298	94
30	65	10	65	56	318,9478887	131
30	80	10	80	79	392,5512476	169
40	20	10	20	16	106,4385619	46
40	35	10	35	36	186,2674833	81
40	50	10	50	58	266,0964047	118
40	65	10	65	72	345,9253262	147
40	80	10	80	54	425,7542476	144
50	20	9	15	24	112,8771238	48
50	35	10	35	36	197,5349666	81
50	50	10	50	38	282,1928095	98
50	65	10	65	44	366,8506523	119
50	80	10	80	54	451,5084952	144

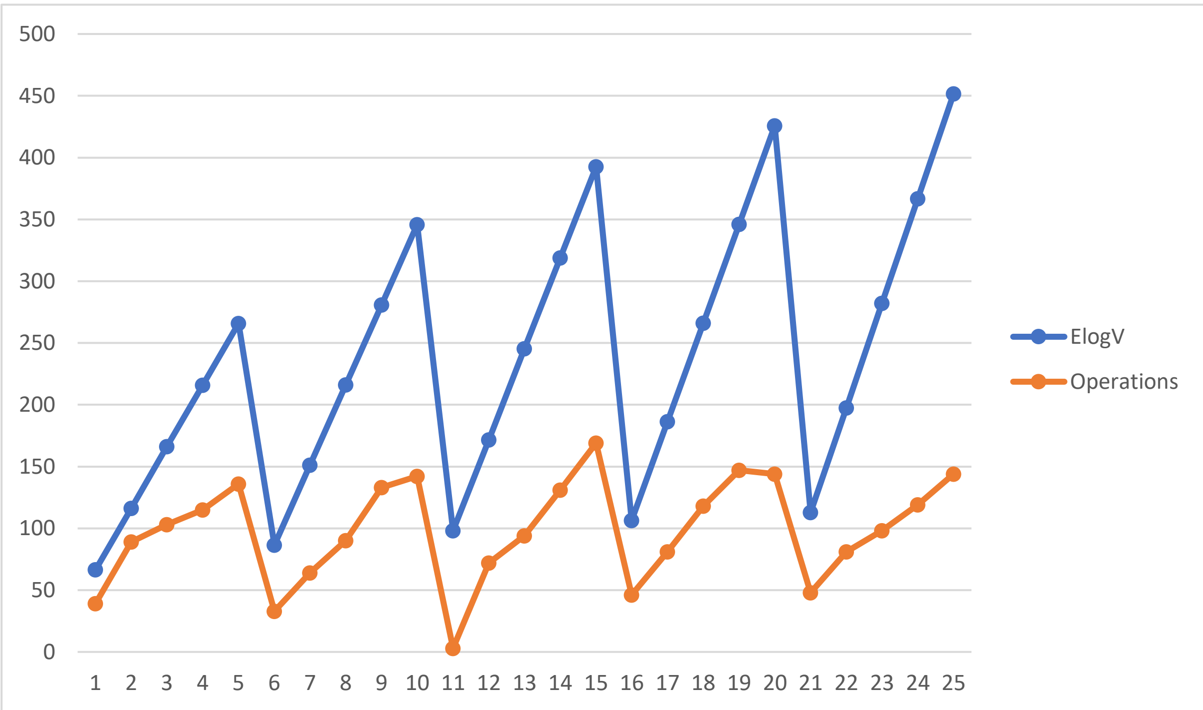
Graph 1: To show the relationship between Operations and the Vertices (V)



Graph 2: To show the relationship between Operations and the Edges (E)



Graph 3: To show the relationship between Operations (blue) and E log V (orange)

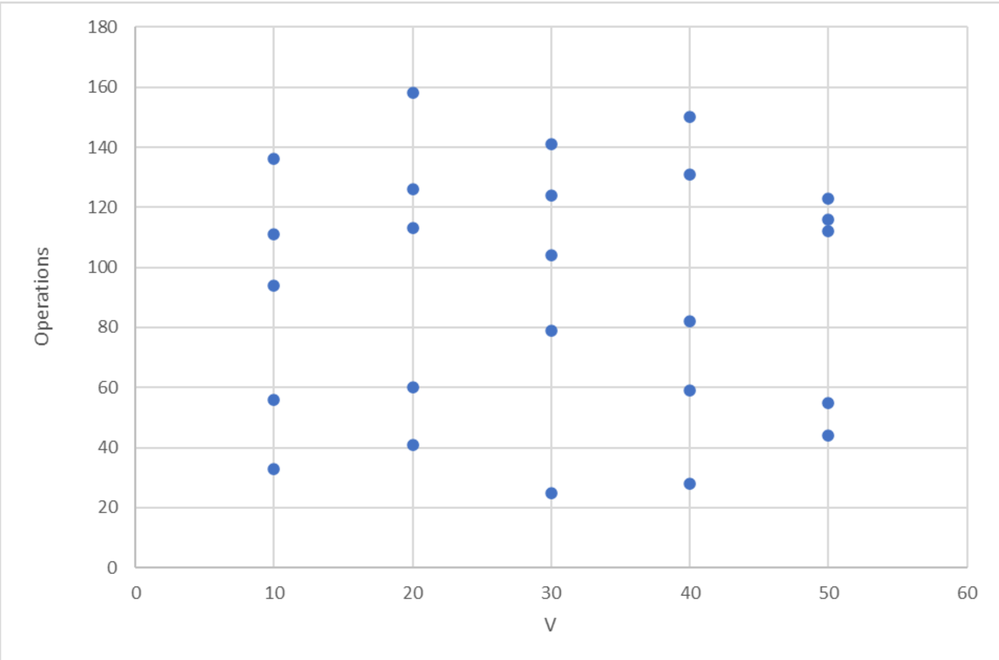


- <u>Results of very small values of edges with instrumentation that only increments every time a comparison is performed</u>

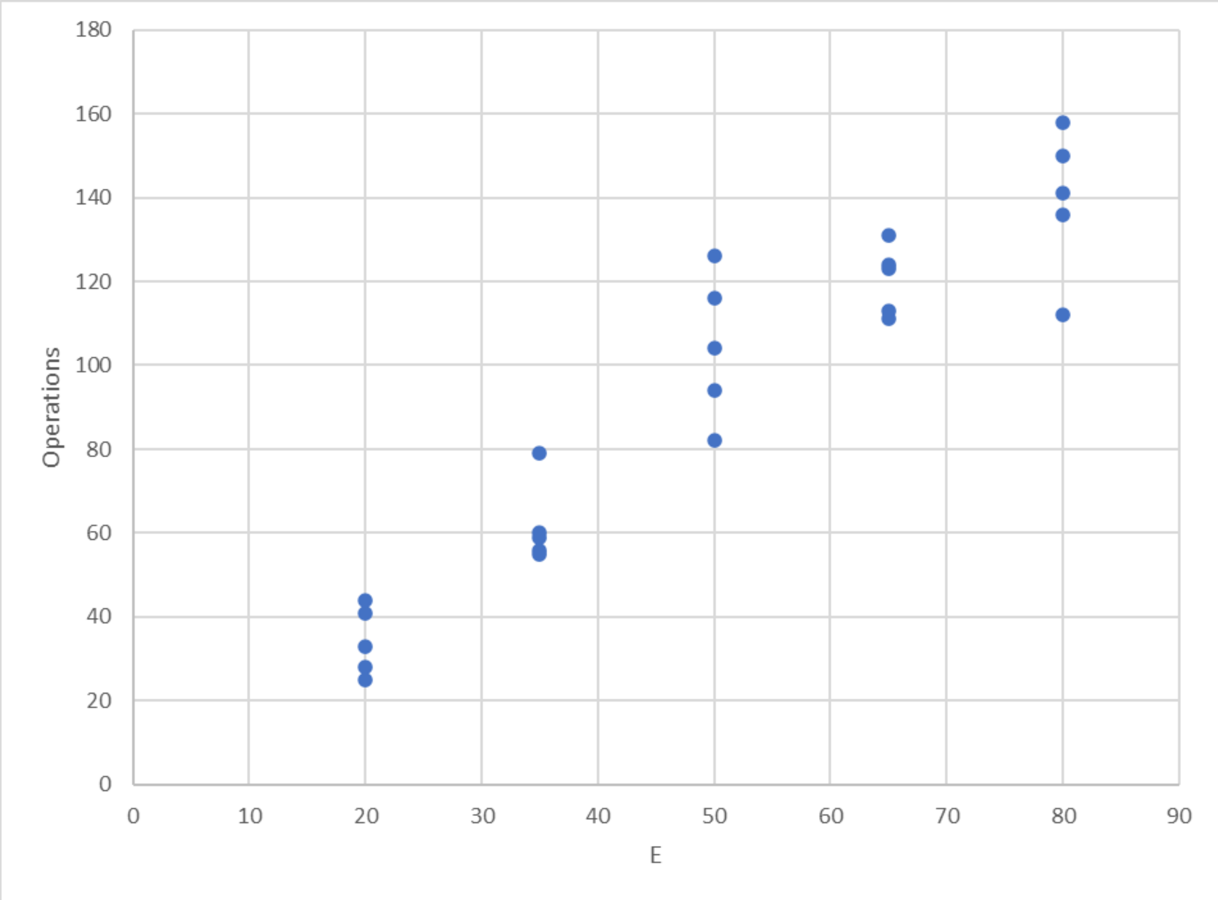
Table 1: To show the values of the data that is being used in the experiment

V	E	Vc	Ec	Pc	ElogV	Operations
10	20	33	0	0	66,43856	33
10	35	56	0	0	116,2675	56
10	50	94	0	0	166,0964	94
10	65	111	0	0	215,9253	111
10	80	136	0	0	265,7542	136
20	20	41	0	0	86,43856	41
20	35	60	0	0	151,2675	60
20	50	126	0	0	216,0964	126
20	65	113	0	0	280,9253	113
20	80	158	0	0	345,7542	158
30	20	25	0	0	98,13781	25
30	35	79	0	0	171,7412	79
30	50	104	0	0	245,3445	104
30	65	124	0	0	318,9479	124
30	80	141	0	0	392,5512	141
40	20	28	0	0	106,4386	28
40	35	59	0	0	186,2675	59
40	50	82	0	0	266,0964	82
40	65	131	0	0	345,9253	131
40	80	150	0	0	425,7542	150
50	20	44	0	0	112,8771	44
50	35	55	0	0	197,535	55
50	50	116	0	0	282,1928	116
50	65	123	0	0	366,8507	123
50	80	112	0	0	451,5085	112

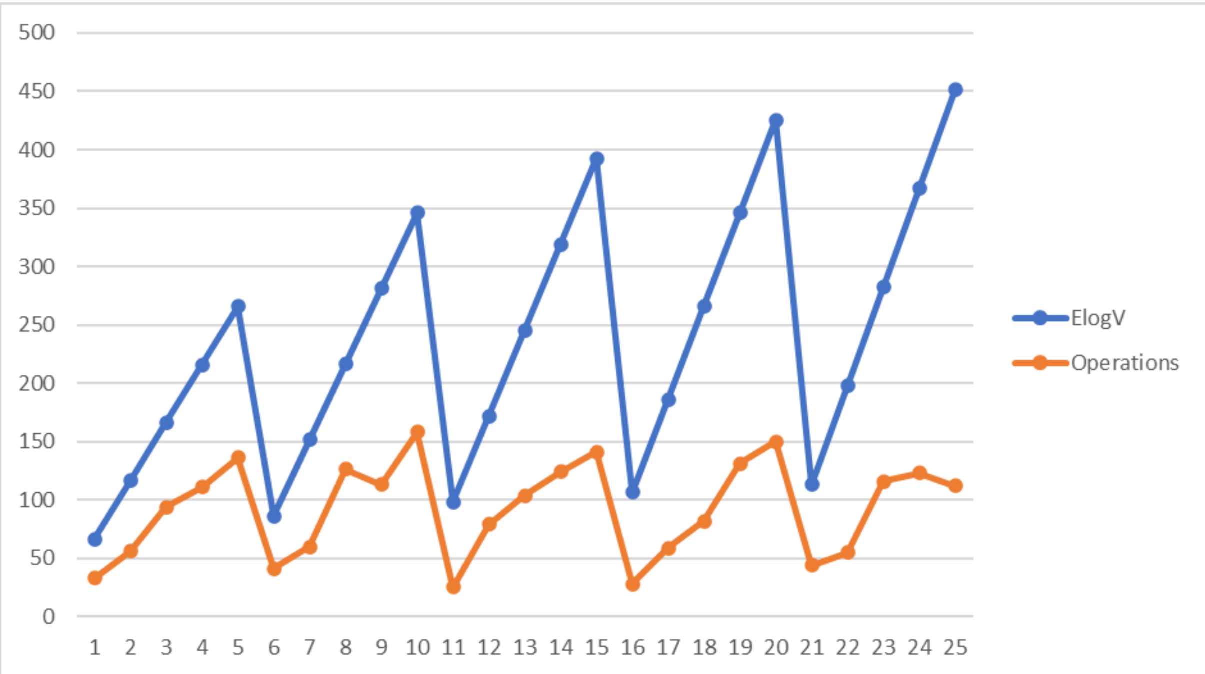
Graph 1: To show the relationship between Operations and the Vertices (V)

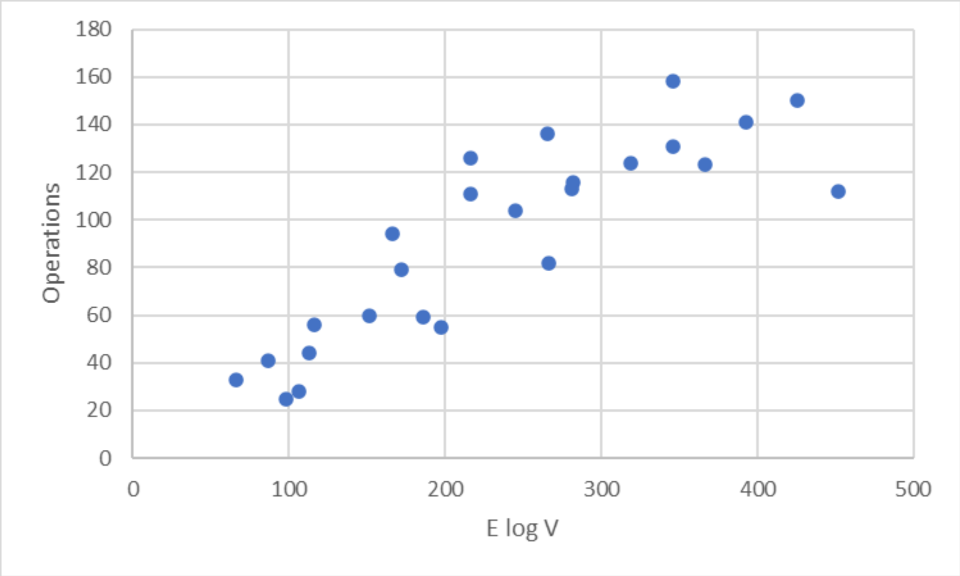


Graph 2: To show the relationship between Operations and the Edges (E)



Graph 3 & 4: To show the relationship between Operations (blue) and E log V (orange)





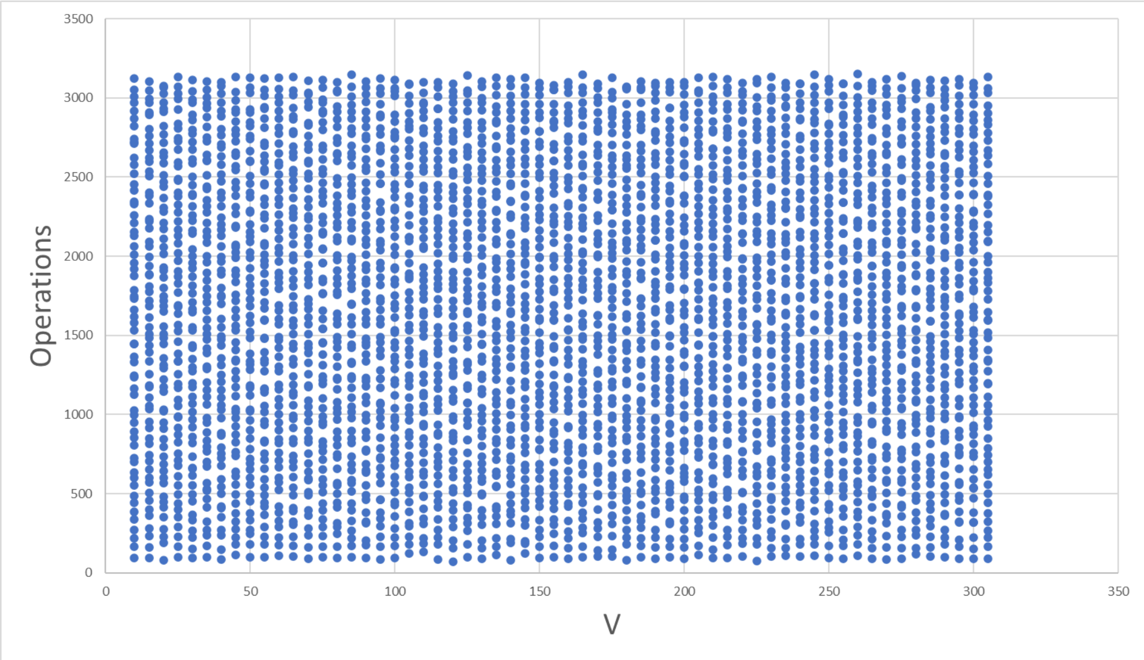
- Results of very large values of edges with instrumentation that only increments every time a comparison is performed

Table 1: To show the values of the data that is being used in the experiment

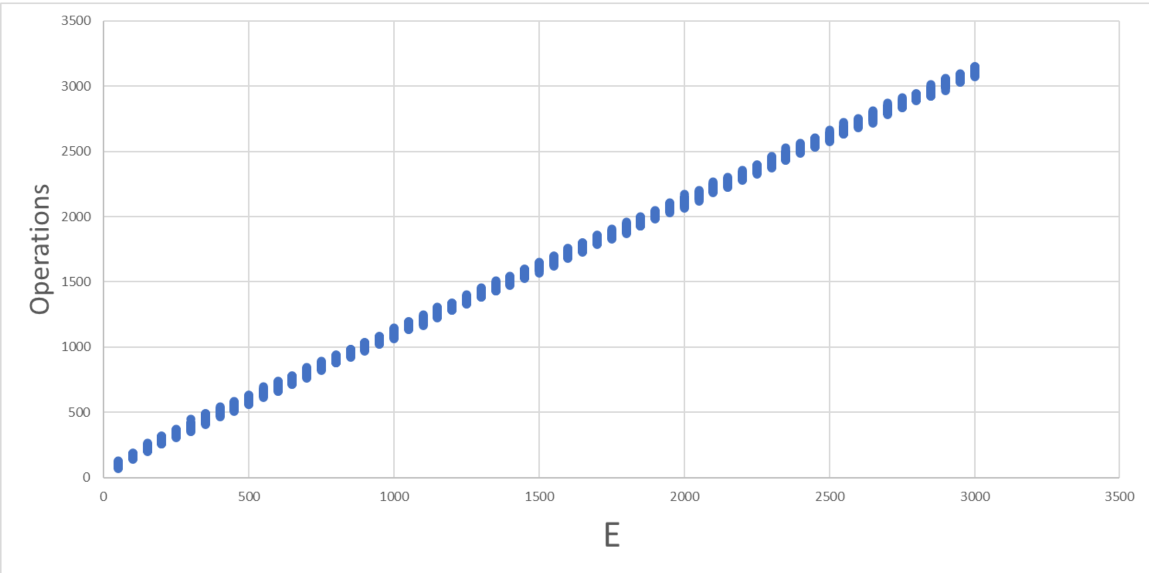
V	E	Vc	Ec	Pc	E log V	Operations
10	50	96	0	0	166,0964047	96
10	100	164	0	0	332,1928095	164
10	150	220	0	0	498,2892142	220
10	200	273	0	0	664,385619	273
10	250	339	0	0	830,4820237	339
10	300	384	0	0	996,5784285	384
10	350	442	0	0	1162,674833	442
10	400	486	0	0	1328,771238	486
10	450	554	0	0	1494,867643	554
10	500	599	0	0	1660,964047	599
10	550	634	0	0	1827,060452	634
10	600	699	0	0	1993,156857	699
10	650	726	0	0	2159,253262	726
10	700	804	0	0	2325,349666	804
10	750	852	0	0	2491,446071	852
10	800	891	0	0	2657,542476	891
10	850	949	0	0	2823,638881	949
10	900	996	0	0	2989,735285	996
10	950	1024	0	0	3155,83169	1024
10	1000	1125	0	0	3321,928095	1125
10	1050	1162	0	0	3488,0245	1162
10	1100	1234	0	0	3654,120904	1234
10	1150	1246	0	0	3820,217309	1246
10	1200	1330	0	0	3986,313714	1330
10	1250	1365	0	0	4152,410119	1365
10	1300	1446	0	0	4318,506523	1446
10	1350	1444	0	0	4484,602928	1444
10	1400	1530	0	0	4650,699333	1530
10	1450	1580	0	0	4816,795738	1580
10	1500	1616	0	0	4982,892142	1616
10	1550	1659	0	0	5148,988547	1659
10	1600	1731	0	0	5315,084952	1731
10	1650	1745	0	0	5481,181357	1745
10	1700	1787	0	0	5647,277761	1787
10	1750	1876	0	0	5813,374166	1876

Values do continue but too many to fit in document

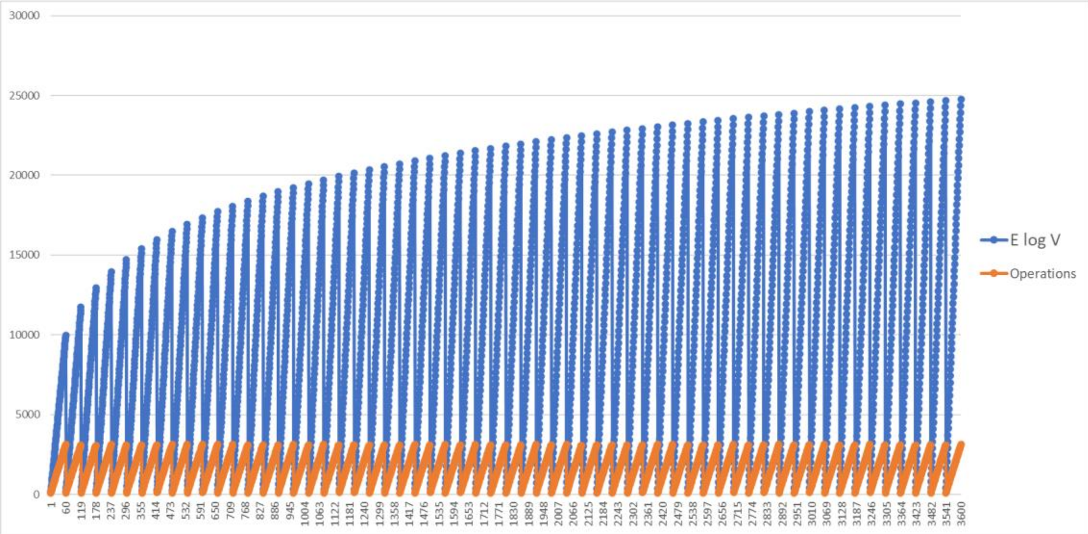
Graph 1: To show the relationship between Operations and the Vertices (V)

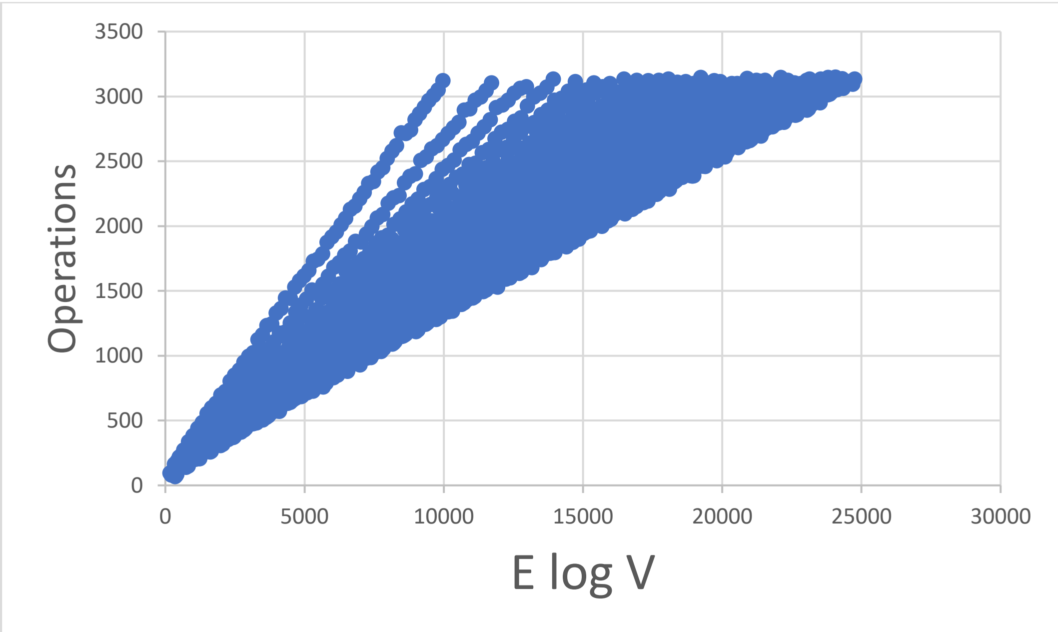


Graph 2: To show the relationship between Operations and the Edges (E)



Graph 3 & 4: To show the relationship between Operations and E log V Values





4. **Discuss what the results mean. Compare your results to the theoretical bounds.**
   The results that I have concluded and produced using my program depict that the theoretical bounds do hold to be true however as there is a lot of variability throughout the conducting of these graphs and the experiments the lines produced in the final graph 4 has outliers such as the first line. However the theoretical bound of E log V does hold true as seen in Graph 3 on page 7 there is a bound that limits the operations to pass through showing true to our experiments and the goal of the experiment that the algorithm does fit the performance of the theoretical bound.
5. **Summary statistics from your use of git to demonstrate usage**
   0: commit b201f72528eab606d90cf2b598ed4d5365a3ba9e
   1: Author: Matthew Schramm <mattschramm1235@gmail.com>
   2: Date: Fri May 5 12:32:29 2023 +0200
   3:

4: Final and complete version ready to hand in

5:

6: commit cea04109e5eb070bf613ffa5504f6715a0e980f3

7: Author: Matthew Schramm <mattschramm1235@gmail.com>

8: Date: Thu May 4 16:04:37 2023 +0200

9:

...

10: code working and files generated