**Report Dijkstra Algorithm**

Throughout the program, vectors were heavily used in order to utilize a matrix where needed. For example for storing the distance from one vertex to another if an edge between them exists which is checked by the adjacent function. Furthermore the path matrix was used to store information about the parents and keep track of going from vertex x to y by going from x to a, a to b and then b to y. The priority queue was implemented with heaps as usually done before.

In this implementation I used the Graph and Priority Queue classes, but I was not able to make effective use of the Priority class due to my lack of experience and never having heard graph theory before. Additionally, instead of making use of the Dijkstra by having it as a child class of shortest path, shortest path was implemented as one class by adding the two extra functions for calculating the path and the path size.

Additionally the monte carlo simulation was done with a function called monte\_carlo and was put into the Priority Queue class for better readability of main.

The results for 20% and 40% were 6.07 and 4.70 respectively we should be able to see it approximate 5 as we increase the number of iterations and the density.