CS519 Design, Analysis, and Implementation of Algorithms
Fall 2019 Homework 6
Aashish Adhikari

1.Implemented

2.(1 point) Compare the running times of the two algorithms as a function of the graph sizes (number of edges). Show the running times a table and a plot in the report.

Solution
Note: I emailed you regarding what the plot is going to be and I assume this is what you are looking for. Blue data is for Prim's Algorithm.

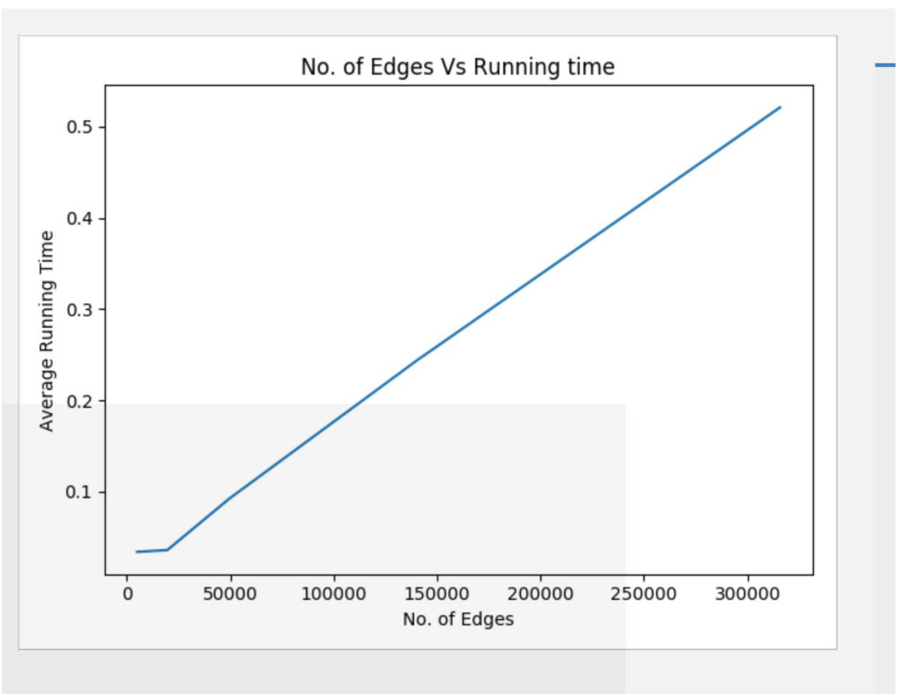
No of edges	*Corresponding Running Times*	*Corresponding Running Times*
4994	*0.034*	*0.022*
19607	*0.036*	*0.085*
49370	*0.092*	*0.255*
140359	*0.244*	*0.821*
315745	*0.521*	*1.987*

For Kruskal's Algorithm:
No of edges= [4994, 19607, 49370, 140359, 315745]
Corresponding Running Times = [0.034, 0.036, 0.092, 0.244, 0.521]

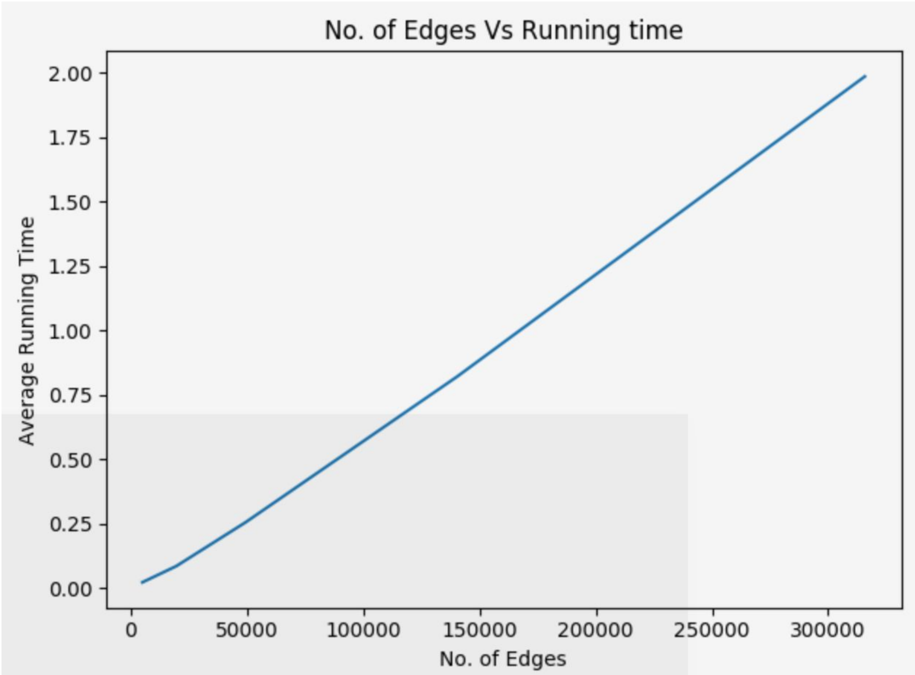
For Prim's Algorithm:
No of edges =[4994, 19607, 49370, 140359, 315745]
Corresponding Running Times = [0.022, 0.085, 0.255, 0.821, 1.987]

As seen above, in one of the cases Prim's algorithm achieved a smaller running time. Else, Kruskal's algorithm beat Prim's algorithm on every test case.

Following is the plot for Kruskel's no of edges vs running time. The number of edges used are from the assignment webpage. Please note that the webpage mentions that the odd test cases are for Prim while it is the opposite when tested on server.



Following is the plot for Prim's Algorithm.



3.(1 point) Let 'maximum spanning tree' be defined as a spanning tree with the maximum total weight. Define the *cut property* for maximum spanning tree as follows. Suppose *X* is a set of edges in a maximum spanning tree. Choose a set of vertices *S* such that no edges in *X* cross from nodes in *S* to nodes in *V-S*. Let *e* be the heaviest edge not in *X* that crosses from *S* to *V-S*. Show that *X union {e}* is a subset of a maximum spanning tree.

Solution:

Case 1: If e is already a part of the maximum cost spanning tree, X union {e} is a subset of the maximum cost spanning tree, T.

Case 2: Assume there is another edge e' that is in the maximum cost spanning tree T but e does not belong to the maximum cost spanning tree. (Because MST should be connected) If we add e to T, there are 2 paths between any two nodes in T. Thus makes T cyclic. If we remove e', then

T' = T union {e} - {e'}

This removes the cycle. This T' is connected and acyclic.

Now, all we need to do is show that this T' is a maximum spanning tree.

Say, Weight of T' = W(T') and Weight of T = W(T)

Therefore,

W(T')
= W(T) + W(e) - W(e')

Since we know that W(e) >= W(e'), hence W(e) - W(e') >= 0

Hence, weight of T' >= weight of T
Hence, T' is a maximum spanning tree and X union {e} is a subset of a maximum spanning tree.

4.(1 point) A barber shop serves *n* customers in a queue. They have service times *t1,...tn*. Only one customer can be served at any time. The waiting time for any customer is the sum of the service times of all previous customers. How would you order the customers so that the total waiting time for all customers will be minimized? Carefully justify your answer.

Solution:
If we want to minimize the total waiting time for all customers, this means that the average waiting time of a customer should be minimum. This can be achieved only if we sort the customers with ascending service times.

Consider a case to two customers. Say A is the customer with shorter service time and B is the customer with the longer service time. If A is served before the B, then A has to wait only their own service time, while B has to wait their own service time plus the service time of A . However, if we serve B first, B has to wait only their own service time but A has to wait a total of the service time B plus their own service time. Hence the total wait time T = 2 * service time of B+ service time of A instead of 2 * service time of A + service time of B. So the average waiting time Avg = T/2 is less if we schedule A first.

Extending this to 3 customers, Avg = T/3 and the idea follows. Hence, to minimize the waiting time for all customers on average, we need to sort the customers in ascending order of service time.