Algorithms and Data Structures

ITCS 6114

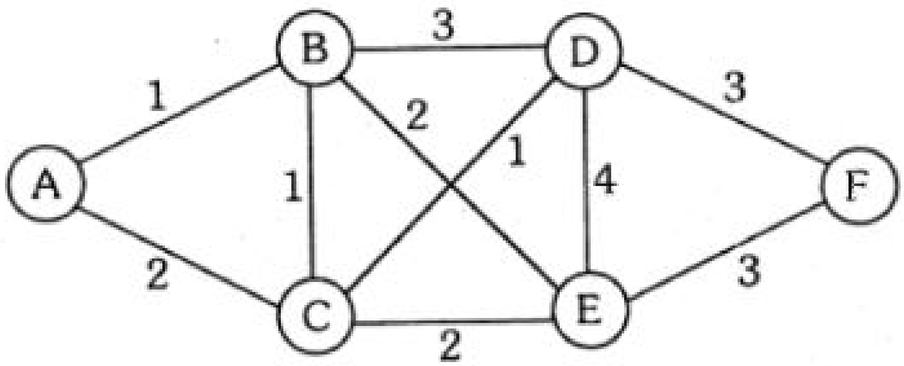
Dr. Dewan Ahmed

Project 2 Report

Submitted by

Darshan Shah

(1) Input Graph 1 -



6	10	U
A	B	1
A	C	2
B	C	1
B	D	3
B	E	2
C	D	1
C	E	2
D	E	4
D	F	3
E	F	3
A		

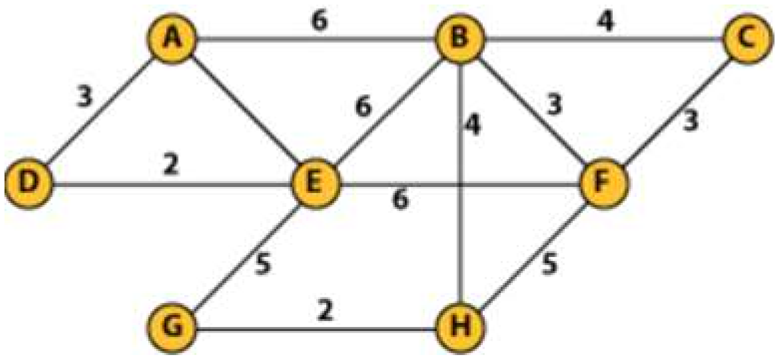
Dijkstra’s Output-

```
Running time in Microseconds : 16

Nodes from Source(A)            Path                    Minimum Distance
B                               B-                              1
C                               C-                              2
D                               C-D-                            3
E                               B-E-                            3
F                               C-D-F-                          6
```

Prims Output -

```
Running time in Microseconds : 16

The Minimum Spanning Tree using Prim's Algorithm:
A -> B (1)
B -> C (1)
C -> D (1)
B -> E (2)
D -> F (3)
Minimum MST Cost: 8
```

(2) Input Graph 2



8	13	U
A	B	6
A	D	3
A	E	8
B	C	4
B	E	6
B	F	3
B	H	4
C	F	3
D	E	2
E	F	6
E	G	5
F	H	5
G	H	2
A		

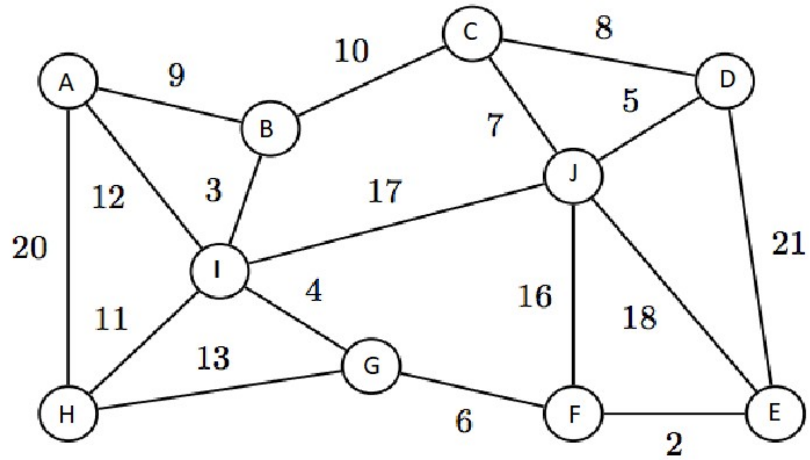
Dijkstra’s Output-

```
Running time in Microseconds : 19

Nodes from Source(A)            Path                    Minimum Distance
B                               B-                              6
C                               B-C-                            10
D                               D-                              3
E                               D-E-                            5
F                               B-F-                            9
G                               D-E-G-                          10
H                               B-H-                            10
```

Prims Output-

```
Running time in Microseconds : 27

The Minimum Spanning Tree using Prim's Algorithm:
H -> B (4)
F -> C (3)
A -> D (3)
D -> E (2)
B -> F (3)
E -> G (5)
G -> H (2)
Minimum MST Cost: 22
```

(3) Input Graph 3



12	21	U
A	B	1306
A	E	2661
A	F	2161
B	D	919
B	C	629
C	D	435
D	F	1225
D	G	1983
E	F	1483
E	G	1532
E	H	661
F	G	1258
G	K	2113
G	L	2161
H	K	1145
H	I	1613
I	J	338
I	K	725
J	K	383
J	L	2145
K	L	1709
A		

Dijkstra’s Output-

```
Running time in Microseconds : 26

Nodes from Source(A)            Path                    Minimum Distance
B                               B-                              9
C                               B-C-                            19
D                               B-C-D-                          27
E                               I-G-F-E-                                24
F                               I-G-F-                          22
G                               I-G-                            16
H                               H-                              20
I                               I-                              12
J                               B-C-J-                          26
```

Prims Output -

```
Running time in Microseconds : 36

The Minimum Spanning Tree using Prim's Algorithm:
A -> B (9)
B -> C (10)
J -> D (5)
F -> E (2)
G -> F (6)
I -> G (4)
I -> H (11)
B -> I (3)
C -> J (7)
Minimum MST Cost: 57
```

Source Code Files

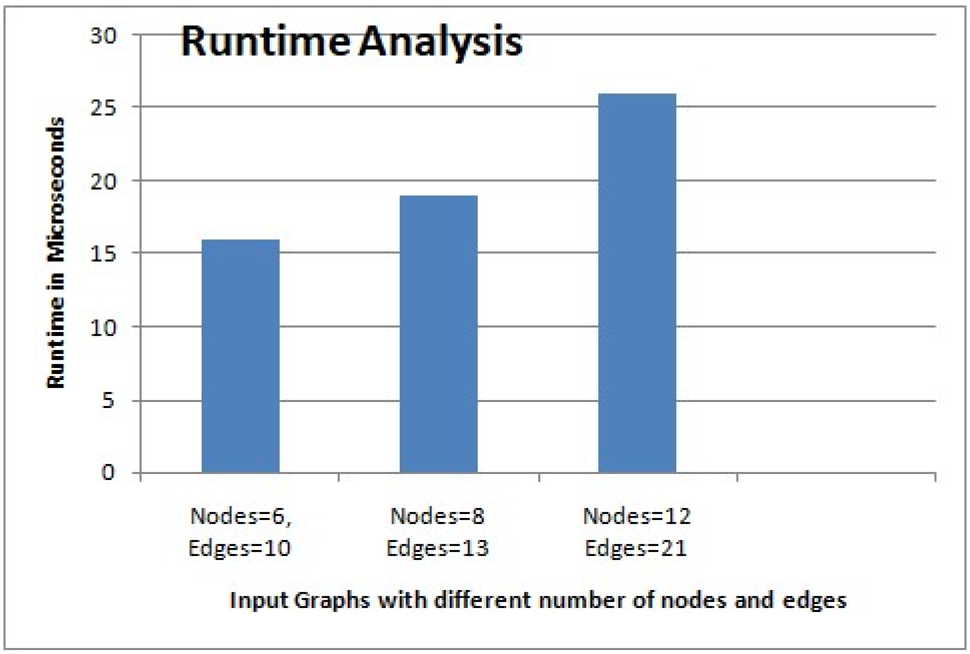
Darshan_Shah_Sour
ce_Code_and_Input_

Runtime Analysis

Dijkstra Algorithm-Shortest Path

After performing big-oh analysis of Dijkstra's algorithm, runtime of the algorithm seems to depend on the number of vertices in the input graph. In general, the runtime seemed to increase as the number of input vertices increased as shown in the graph below.

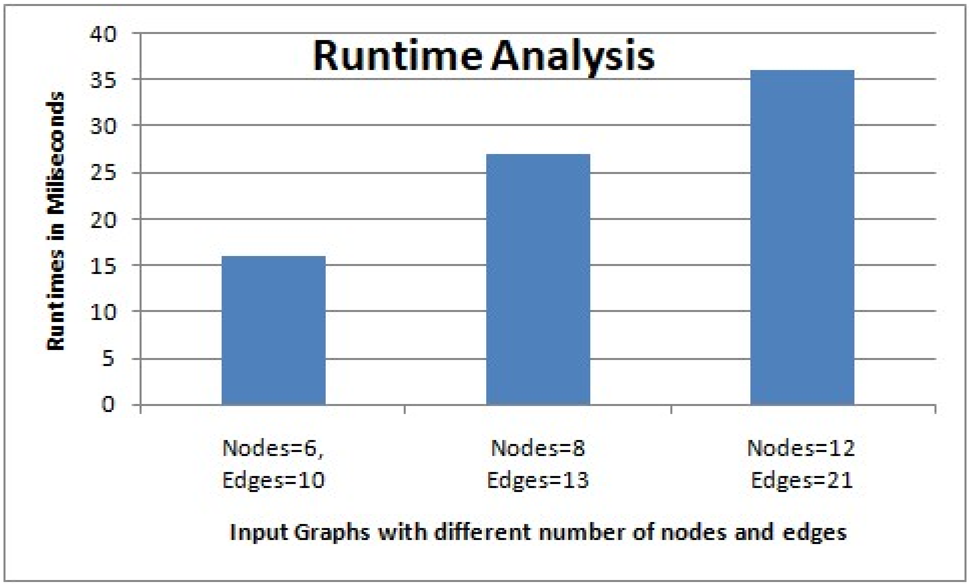
This makes sense as Dijkstra's algorithm runs in O(E log V). In other words, the runtime of Dijkstra is directly proportional to the number of edges and vertices in a given input graph. The graph's upward slope depicts this proportionality.



Prim's Algorithm-Minimum Spanning Tree

Performing big-oh analysis of Prim's Shortest path results in runtime of O(E log V), similar to Dijkstra's shortest path algorithm above. This is because Prim's algorithm adds one vertex/edge at one time to the minimum spanning tree and at each iteration, removes the minimum edge from heap and updating the distances of adjacent vertices. Each edge will be updated at most twice (once from each endpoint). Each heap operation takes log V time as there are V nodes in heap. As a result, the final runtime for Prim's algorithm is O(2*E log V) or O(E log V).

The following graph shows the runtimes of Prim's algorithm over a few graphs (discussed above). As seen below, the runtime increases as the number of nodes and edges increases. This reflects the expected, theoretical runtime of O(E log V).


Runtime Analysis
Runtimes in Miliseconds
40
35
30
25
20
15
10
5
0
Nodes=6,
Edges=10
Nodes=8
Edges=13
Nodes=12
Edges=21
Input Graphs with different number of nodes and edges