

```
D:\Java\jdk1.8.0_102\bin\java.exe "-javaagent:D:\IntelliJ IDEA Community
Edition 2019.2\lib\idea_rt.jar=60125:D:\IntelliJ IDEA Community Edition 2019
.2\bin" -Dfile.encoding=UTF-8 -classpath "D:\Java\jdk1.8.0_102\jre\lib\
charsets.jar;D:\Java\jdk1.8.0_102\jre\lib\deploy.jar;D:\Java\jdk1.8.0_102\
jre\lib\ext\access-bridge-32.jar;D:\Java\jdk1.8.0_102\jre\lib\ext\cldrdata.
jar;D:\Java\jdk1.8.0_102\jre\lib\ext\dnsns.jar;D:\Java\jdk1.8.0_102\jre\lib\
ext\jaccess.jar;D:\Java\jdk1.8.0_102\jre\lib\ext\jfxrt.jar;D:\Java\jdk1.8.
0_102\jre\lib\ext\localedata.jar;D:\Java\jdk1.8.0_102\jre\lib\ext\nashorn.
jar;D:\Java\jdk1.8.0_102\jre\lib\ext\sunec.jar;D:\Java\jdk1.8.0_102\jre\lib\
ext\sunjce_provider.jar;D:\Java\jdk1.8.0_102\jre\lib\ext\sunmscapi.jar;D:\
Java\jdk1.8.0_102\jre\lib\ext\sunpkcs11.jar;D:\Java\jdk1.8.0_102\jre\lib\ext
\zipfs.jar;D:\Java\jdk1.8.0_102\jre\lib\javaws.jar;D:\Java\jdk1.8.0_102\jre\
lib\jce.jar;D:\Java\jdk1.8.0_102\jre\lib\jfr.jar;D:\Java\jdk1.8.0_102\jre\
lib\jfxswt.jar;D:\Java\jdk1.8.0_102\jre\lib\jsse.jar;D:\Java\jdk1.8.0_102\
jre\lib\management-agent.jar;D:\Java\jdk1.8.0_102\jre\lib\plugin.jar;D:\Java
\jdk1.8.0_102\jre\lib\resources.jar;D:\Java\jdk1.8.0_102\jre\lib\rt.jar;C:\
Users\Ankit Pandita\Desktop\Project2_GraphAlgo\out\production\
Project2_GraphAlgo" edu.uncc.cci.algods.Main
Select the input file from 1, 2, 3, 4, 5 or 6.
Note: File 1, 2, 3 and 4 have Undirected Graphs and file 5 and 6 have
Directed Graphs.
```

1

Choose an action:

1. Display Shortest Path (using Dijkstra's Algorithm)
2. Display Minimal Spanning Tree (using Kruskal's Algorithm)
3. Exit

1

Number of vertices = 9

Number of edges = 14

Selected Graph is Undirected.

Applying Dijkstra's Algorithm:

	A	B	C	D	E	F	G	H	I
A	0	3	0	0	0	0	0	7	0
B	3	0	8	0	0	0	0	12	0
C	0	8	0	8	0	5	0	0	3
D	0	0	8	0	8	13	0	0	0
E	0	0	0	8	0	12	0	0	0
F	0	0	5	13	12	0	3	0	0
G	0	0	0	0	0	3	0	1	5
H	7	12	0	0	0	0	1	0	6
I	0	0	3	0	0	0	5	6	0

Source of graph is C

Shortest Path from source:

```
C -> B -> A = 11
C -> B = 8
C -> D = 8
C -> D -> E = 16
C -> F = 5
C -> I -> G = 8
C -> I -> H = 9
C -> I = 3
```

Total time taken = 21504 ns

Choose an action:

1. Display Shortest Path (using Dijkstra's Algorithm)
  2. Display Minimal Spanning Tree (using Kruskal's Algorithm)
  3. Exit
- 2

Number of vertices = 9

Number of edges = 14

Applying Kruskal's Algorithm:

A -> B = 3  
A -> H = 7  
B -> C = 8  
B -> H = 12  
C -> D = 8  
C -> F = 5  
C -> I = 3  
D -> E = 8  
D -> F = 13  
E -> F = 12  
F -> G = 3  
G -> I = 5  
G -> H = 1  
H -> I = 6

Minimum Spanning Tree:

G -> H = 1  
A -> B = 3  
C -> I = 3  
F -> G = 3  
C -> F = 5  
A -> H = 7  
C -> D = 8  
D -> E = 8

Total Cost = 38

Total time taken = 867487 ns

Choose an action:

1. Display Shortest Path (using Dijkstra's Algorithm)
  2. Display Minimal Spanning Tree (using Kruskal's Algorithm)
  3. Exit
- 3

Process finished with exit code 0