MANUAL TO USE THIS PROJECT

This Application basically contains three Runnable files:

1.astar.cpp
2.dijkstra.cpp
3.run.sh

1.Running astar.cpp

Dependencies:
1.astar.hpp
2.GNU/G++ compiler

How to Run :
1.Compile the file in terminal using following command:

Command: g++ astar.cpp

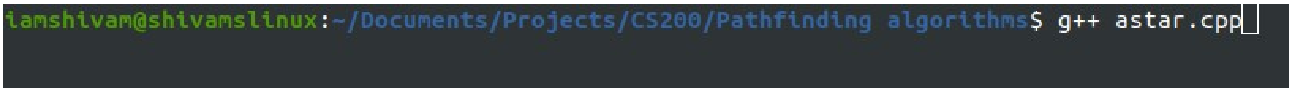
2.Run the file using the following command and please remember to provide the destination node

to which shortest path is to be found as command line input :
Command: ./a.out destinstion_node_id

Ouput :

After the above mentioned steps are followed output will be displayed in the terminal which tells the path length to the destination node and time taken by the algorithm to calculate the path.
Example

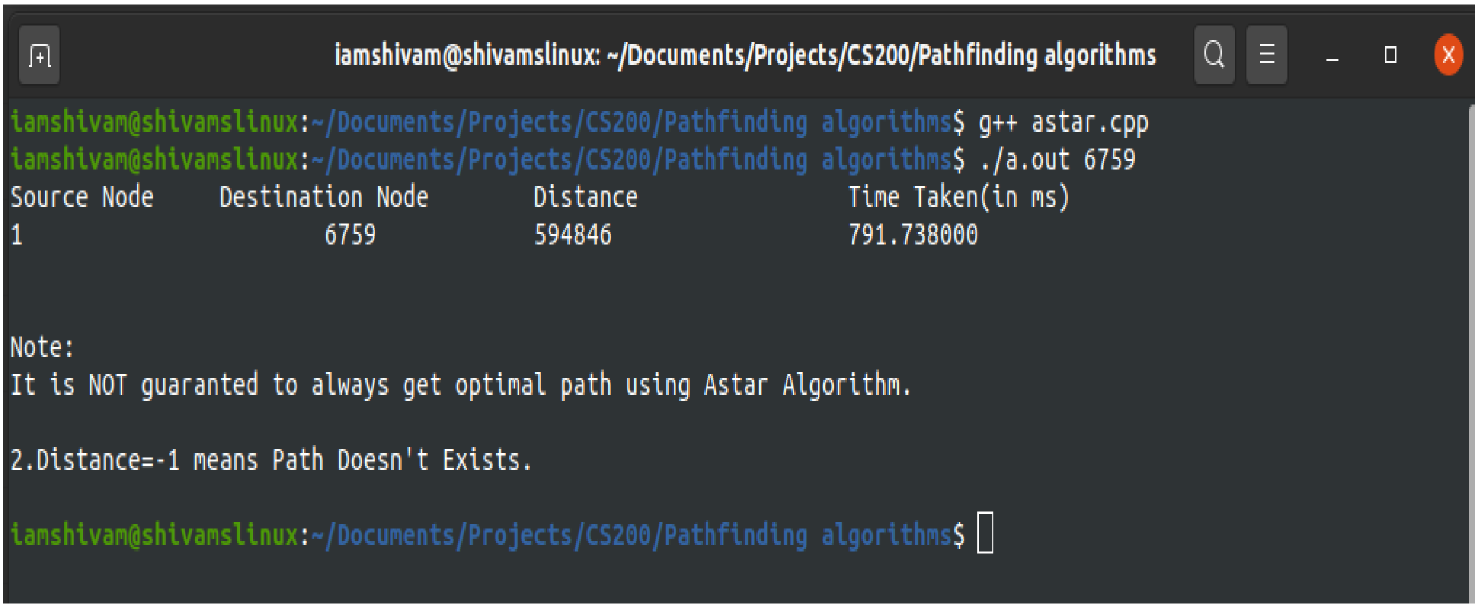
[1.] g++ astar.cpp



```
iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ g++ astar.cpp
```

[2.] ./a.out 6759

3.Output:



iamshivam@shivamslinux: ~/Documents/Projects/CS200/Pathfinding algorithms

```
iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ g++ astar.cpp
iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ ./a.out 6759
Source Node     Destination Node        Distance                Time Taken(in ms)
1                       6759            594846                  791.738000


Note:
It is NOT guaranted to always get optimal path using Astar Algorithm.

2.Distance=-1 means Path Doesn't Exists.

iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ 
```

2.)Running dijkstra.cpp

Dependencies:
1.GNU/G++ compiler
2.OS:Ubunut/Mac OS

How to Run :
1.Compile the file in terminal using following command:

Command: g++ dijkstra.cpp

2.Run the file using the following command and please remember to provide the destination node to which shortest path is to be found as command line input :

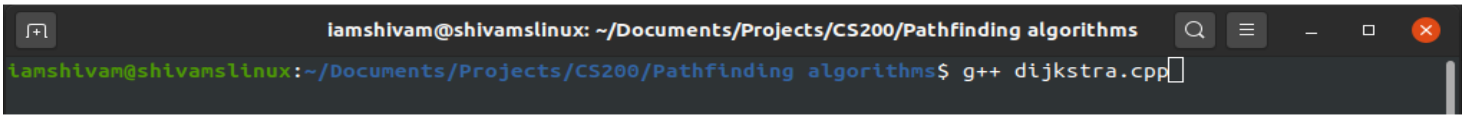
Command: ./a.out destinstion_node_id

Output :

After the above mentioned steps are followed output will be displayed in the terminal which tells the path length to the destination node and time taken by the algorithm to calculate the path.

Example:

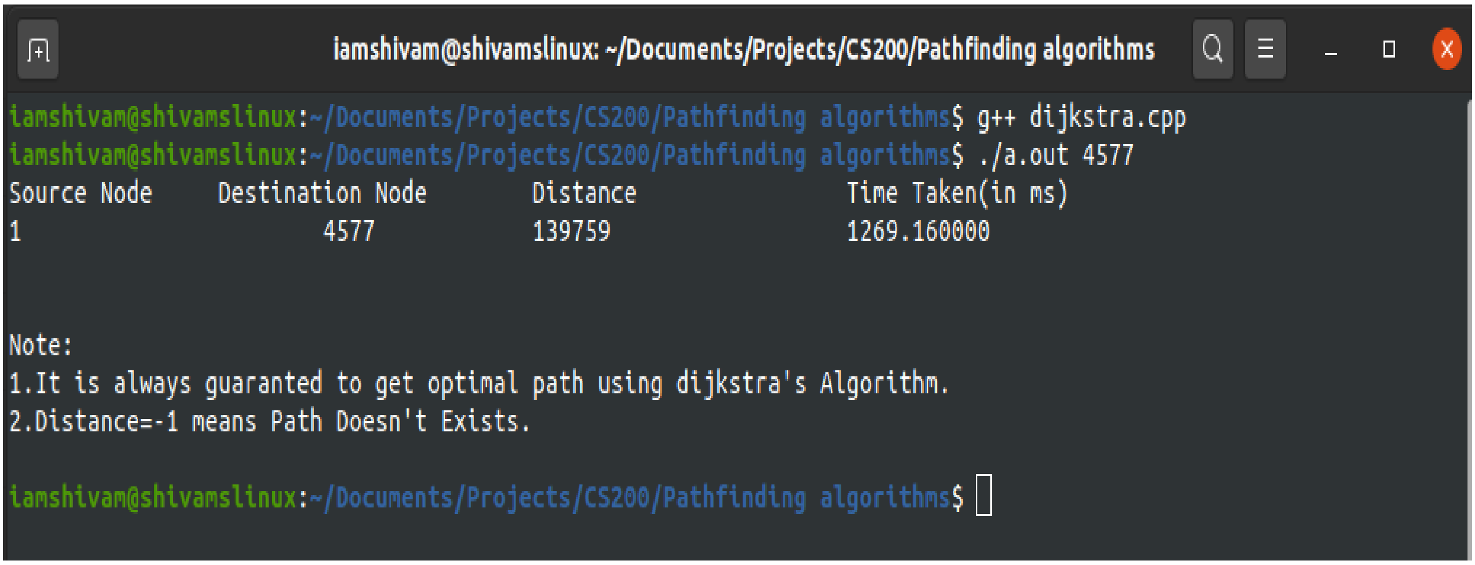
[1.] g++ dijkstra.cpp


iamshivam@shivamslinux: ~/Documents/Projects/CS200/Pathfinding algorithms

```
iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ g++ dijkstra.cpp
```

[2.] ./a.out 4577

[3.]Output:



iamshivam@shivamslinux: ~/Documents/Projects/CS200/Pathfinding algorithms

```
iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ g++ dijkstra.cpp
iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ ./a.out 4577
Source Node     Destination Node        Distance                Time Taken(in ms)
1                       4577            139759                  1269.160000


Note:
1.It is always guaranted to get optimal path using dijkstra's Algorithm.
2.Distance=-1 means Path Doesn't Exists.

iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ 
```

3.)Running run.sh

This file is used to compare dijkstra and astar algorithm over a large dataset of almost 5 lac+ nodes and 20 lac+ edges.This file is used to plot curve between no. Of nodes and time taken by the algorithm to find the shortest path to that specified node.

Dependencies:

1.astar.cpp
2.astar.hpp
3.dijkstra.cpp
4.OS:Ubuntu/Mac OS

How to run :

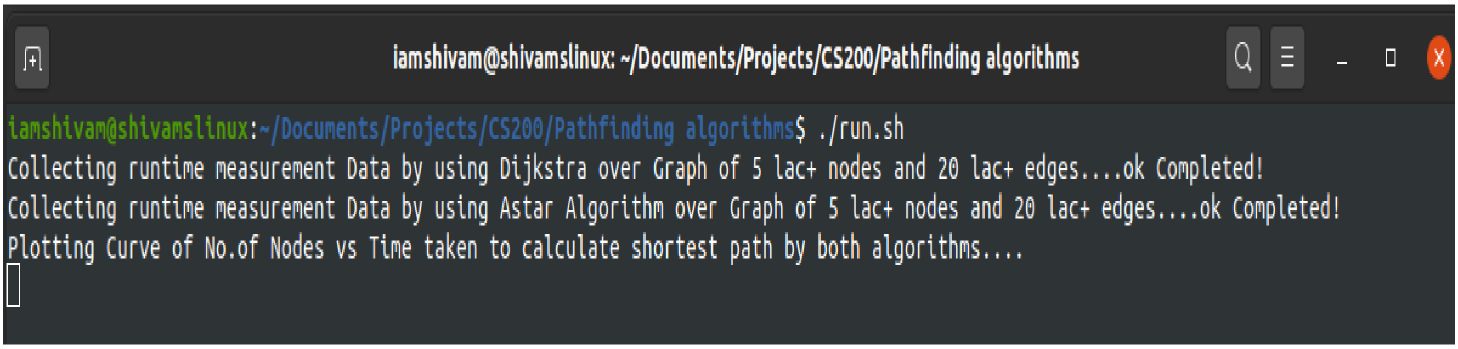
1.Give the permission to the file by using following command:

command: chmod 777 run.sh

2.Run the file by using following command:

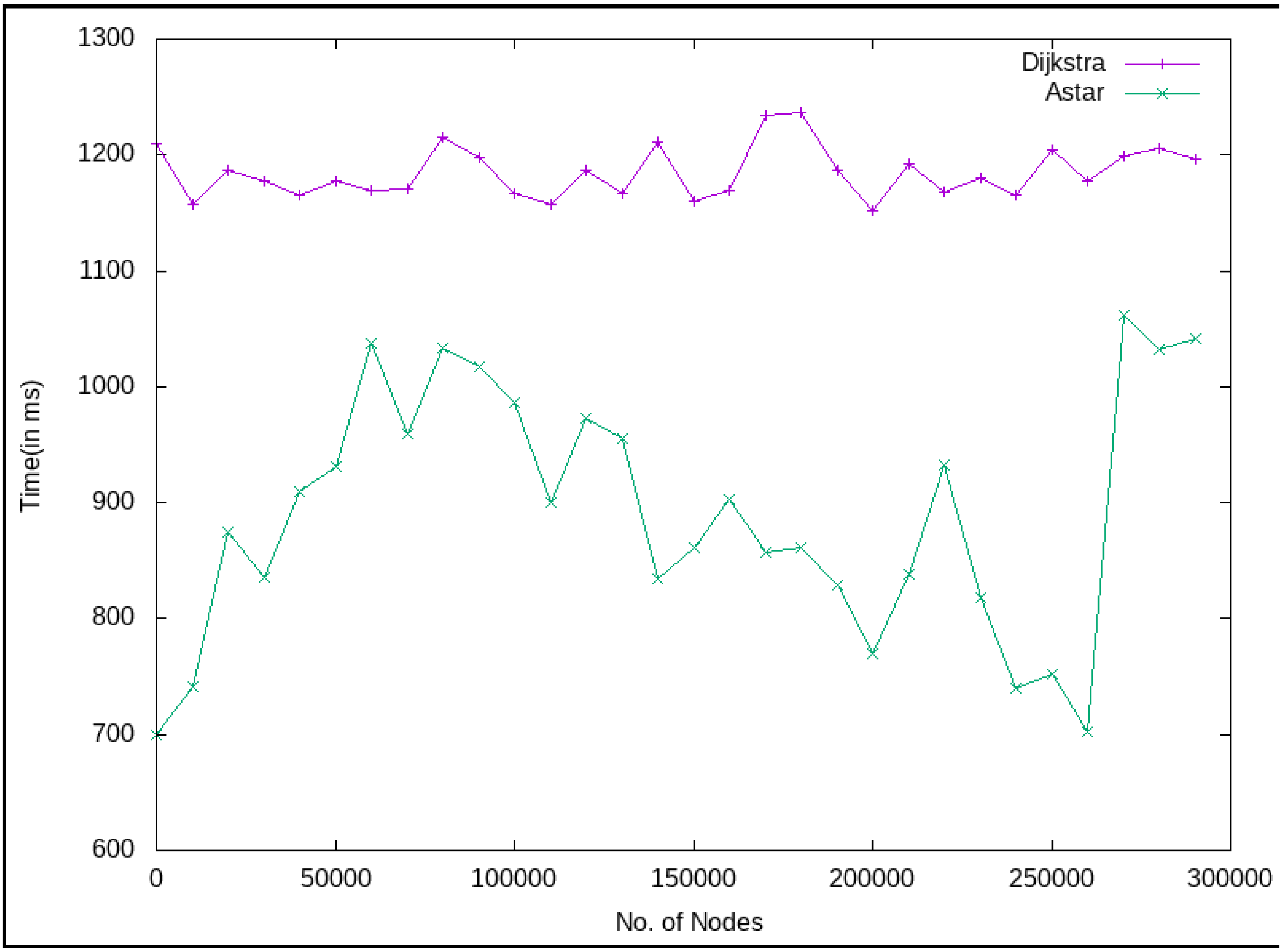
command: ./run.sh

Output:



iamshivam@shivamslinux: ~/Documents/Projects/CS200/Pathfinding algorithms

```
iamshivam@shivamslinux:~/Documents/Projects/CS200/Pathfinding algorithms$ ./run.sh
Collecting runtime measurement Data by using Dijkstra over Graph of 5 lac+ nodes and 20 lac+ edges....ok Completed!
Collecting runtime measurement Data by using Astar Algorithm over Graph of 5 lac+ nodes and 20 lac+ edges....ok Completed!
Plotting Curve of No.of Nodes vs Time taken to calculate shortest path by both algorithms....
```


1300
1200
1100
1000
900
800
700
600
Time(in ms)
0
50000
100000
150000
200000
250000
300000
No. of Nodes
Dijkstra
Astar