

Assignment 2 - Dijkstra

Task a :

In task a , we are asked to modify/create a graph network which can accommodate for edge weights
I have children attributes in Node class to add connected nodes with their respective weights.

```
class Node:
    def __init__(self,name) -> None:
        self.name = name
        self.parent = False
        self.children = []
        self.depth = 0 # depth of node in tree

    def add_child(self,child,weight,depth=0) -> None:
        #self.depth = depth
        self.children.append((weight,child,depth))

node_a.add_child(node_b,2)
node_a.add_child(node_d,6)
```

As shown in the above, I can add child nodes **node_b** and **node_d** to **node_a** with their respective edge weights.

Task b :

In task b , we are asked to implement Dijkstra algorithms for the given graph with nodes and edge connections.

```
node_a = Node('a')
node_b = Node('b')
node_a.add_child(node_b,2)
node_a.add_child(node_d,6)
node_b.add_child(node_c,1)
node_b.add_child(node_e,3)
```

I have created the node and edges as shown above. Connections are added between any two given nodes.
Then we call the Dijkstra algorithm on the graph with a starting and ending node. The output will be either the shortest path existing or No path exists if not possible.

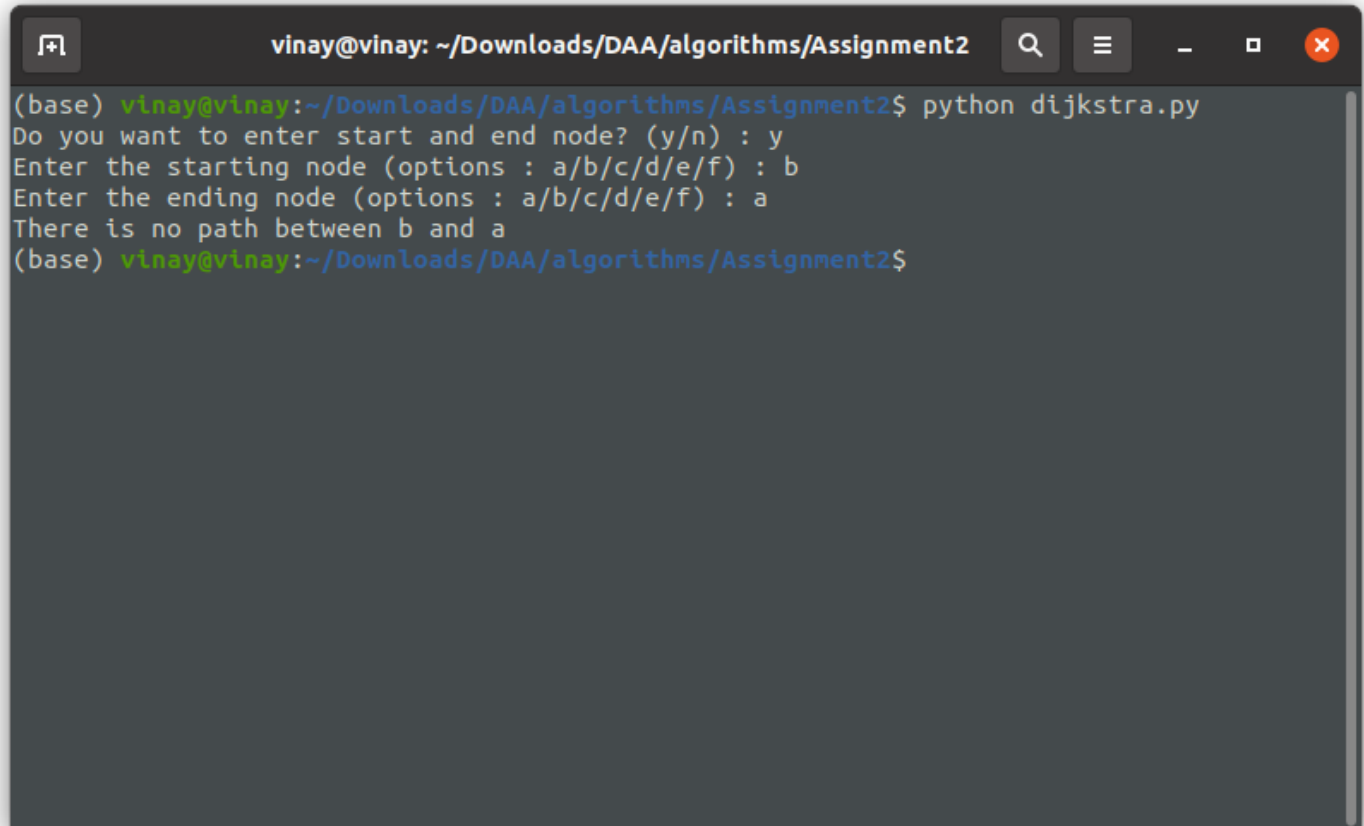
1. Here in the below image, there is provision for user input for start and end nodes. We can choose yes(y)/no(n). If chosen yes(y) then we have to enter start and end nodes.

```
vinay@vinay: ~/Downloads/DAA/algorithms/Assignment2
(base) vinay@vinay:~/Downloads/DAA/algorithms/Assignment2$ python dijkstra.py
Do you want to enter start and end node? (y/n) : y
Enter the starting node (options : a/b/c/d/e/f) : a
Enter the ending node (options : a/b/c/d/e/f) : f
Path between a and f exists
Shortest path : a -> b -> f
Shortest path length : 4
(base) vinay@vinay:~/Downloads/DAA/algorithms/Assignment2$
```

2. If we choose no(n) to not to enter start and end nodes then predefined start and end nodes **a** and **f** are used for searching.

```
vinay@vinay: ~/Downloads/DAA/algorithms/Assignment2
(base) vinay@vinay:~/Downloads/DAA/algorithms/Assignment2$ python dijkstra.py
Do you want to enter start and end node? (y/n) : n
Finding shortest path between a and f
Path between a and f exists
Shortest path : a -> b -> f
Shortest path length : 4
(base) vinay@vinay:~/Downloads/DAA/algorithms/Assignment2$
```

3. If no path exists between start and end nodes, it will report accordingly. Below is an example of that case.



```
vinay@vinay: ~/Downloads/DAA/algorithms/Assignment2
(base) vinay@vinay:~/Downloads/DAA/algorithms/Assignment2$ python dijkstra.py
Do you want to enter start and end node? (y/n) : y
Enter the starting node (options : a/b/c/d/e/f) : b
Enter the ending node (options : a/b/c/d/e/f) : a
There is no path between b and a
(base) vinay@vinay:~/Downloads/DAA/algorithms/Assignment2$
```