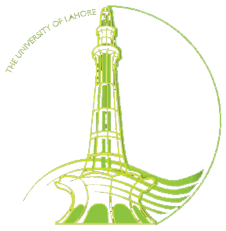


THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

Artificial Intelligence
(CS13217)

Lab Report

Name:	Sameea Naeem
Registration #:	CSU-XS16-139
Lab Report #:	05
Dated:	16-04-2018
Submitted To:	Mr. Usman Ahmed

The University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

Experiment # 5
Implementing Dijkstra's Algorithm

Objective
To understand and implement the Dijkstra's Algorithm

Software Tool
1. OS: Windows
2. IDE: Pycharm
3. Language: Python

1 Theory

Dijkstras algorithm is very similar to Prims algorithm for minimum spanning tree. Like Prims MST, we generate a SPT (shortest path tree) with given source as root. We maintain two sets, one set contains vertices included in shortest path tree, other set includes vertices not yet included in shortest path tree. At every step of the algorithm, we find a vertex which is in the other set (set of not yet included) and has minimum distance from source.

Below are the detailed steps used in Dijkstras algorithm to find the shortest path from a single source vertex to all other vertices in the given graph.
Algorithm
1) Create a set sptSet (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.
2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.
3) While sptSet doesnt include all vertices
.a) Pick a vertex u which is not there in sptSetand has minimum distance value.
.b) Include u to sptSet.
.c) Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less

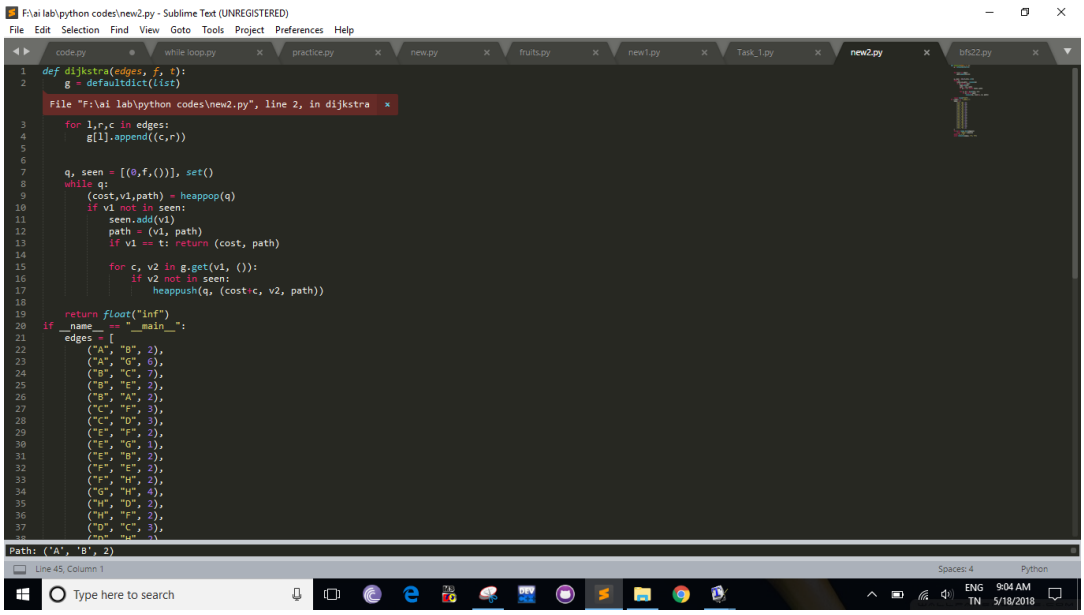
than the distance value of v, then update the distance value of v.

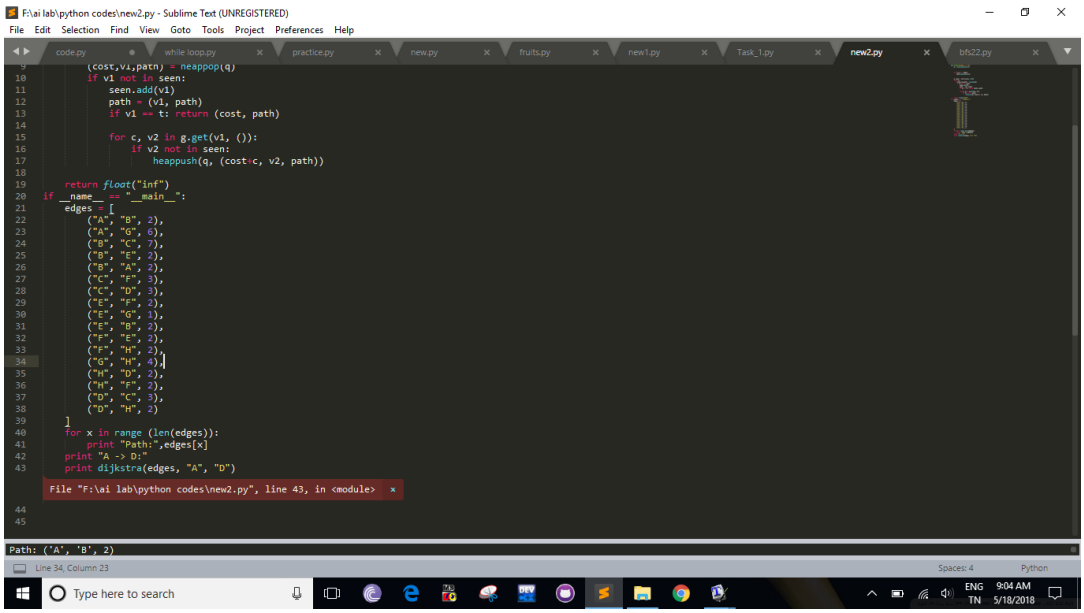
2 Code

```
from collections import defaultdict
from heapq import *

def dijkstra(edges, f, t):
    g = defaultdict(list)
    for l,r,c in edges:
        g[l].append((c,r))


    q, seen = [(0,f,())], set()
    while q:
        (cost,v1,path) = heappop(q)
        if v1 not in seen:
            seen.add(v1)
            path = (v1, path)
            if v1 == t: return (cost, path)

            for c, v2 in g.get(v1, ()):
                if v2 not in seen:
                    heappush(q, (cost+c, v2, path))

    return float("inf")
if __name__ == "__main__":
    edges = [
        ("A", "B", 2),
        ("A", "G", 6),
        ("B", "C", 7),
        ("B", "E", 2),
        ("B", "A", 2),
        ("C", "F", 3),
        ("C", "D", 3),
        ("E", "F", 2),
        ("E", "G", 1),
```

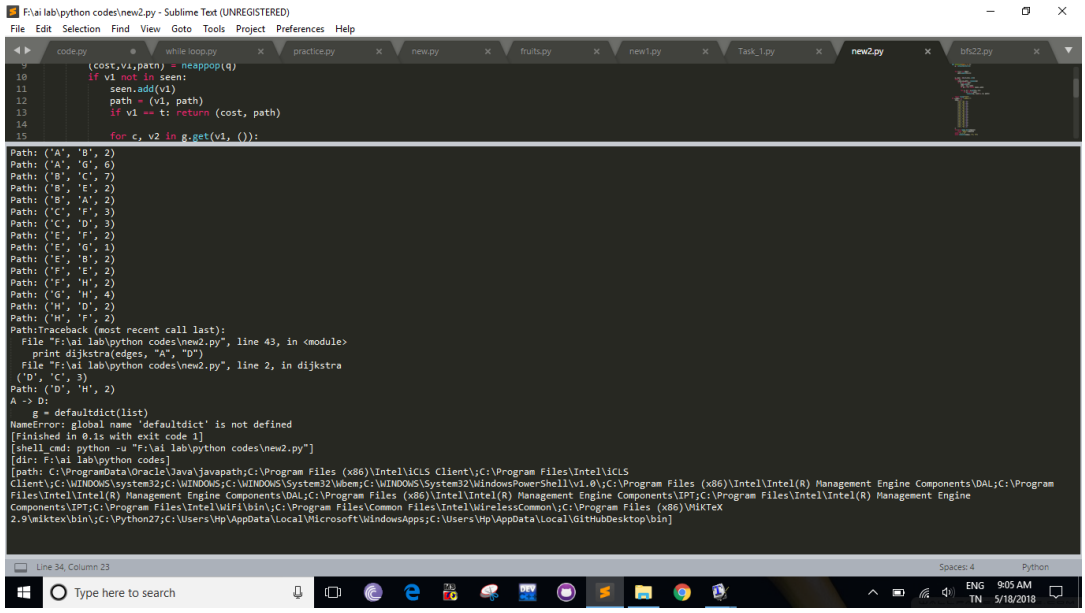
```
            ("E", "B", 2),
            ("F", "E", 2),
            ("F", "H", 2),
            ("G", "H", 4),
            ("H", "D", 2),
            ("H", "F", 2),
            ("D", "C", 3),
            ("D", "H", 2)
        ]
        for x in range(len(edges)):
            print "Path:",edges[x]
        print "A -> D:"
        print dijkstra(edges, "A", "D")

}
```

3 Implement Dijkstra's in Python language:




F:\ai lab\python codes\new2.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
code.py
while loop.py
practice.py
new.py
fruits.py
new1.py
Task_1.py
new2.py
bfs22.py
(cost,v1,path) = heappop(q)
if v1 not in seen:
seen.add(v1)
path = (v1, path)
if v1 == t: return (cost, path)
for c, v2 in g.get(v1, ()):
if v2 not in seen:
heappush(q, (cost+c, v2, path))
return float("inf")
if __name__ == "__main__":
edges = [
("A", "B", 2),
("A", "G", 6),
("B", "C", 7),
("B", "E", 2),
("B", "A", 2),
("C", "F", 3),
("C", "D", 3),
("E", "F", 2),
("E", "G", 1),
("E", "B", 2),
("F", "E", 2),
("F", "H", 2),
("G", "H", 4),
("H", "D", 2),
("H", "F", 2),
("D", "C", 3),
("D", "H", 2)
]
for x in range (len(edges)):
print "Path:",edges[x]
print "A -> D:"
print dijkstra(edges, "A", "D")
File "F:\ai lab\python codes\new2.py", line 43, in <module>
Path: ('A', 'B', 2)
Line 34, Column 23
Spaces: 4
Python
Type here to search
ENG
TN
9:04 AM
5/18/2018

Output:


F:\ai lab\python codes\new2.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
code.py
while loop.py
practice.py
new.py
fruits.py
new1.py
Task_1.py
new2.py
bfs22.py
(cost,v1,path) = heappop(q)
if v1 not in seen:
seen.add(v1)
path = (v1, path)
if v1 == t: return (cost, path)
for c, v2 in g.get(v1, ()):
Path: ('A', 'B', 2)
Path: ('A', 'G', 6)
Path: ('B', 'C', 7)
Path: ('B', 'E', 2)
Path: ('B', 'A', 2)
Path: ('C', 'F', 3)
Path: ('C', 'D', 3)
Path: ('E', 'F', 2)
Path: ('E', 'G', 1)
Path: ('E', 'B', 2)
Path: ('F', 'E', 2)
Path: ('F', 'H', 2)
Path: ('G', 'H', 4)
Path: ('H', 'D', 2)
Path: ('H', 'F', 2)
Path:Traceback (most recent call last):
File "F:\ai lab\python codes\new2.py", line 43, in <module>
print dijkstra(edges, "A", "D")
File "F:\ai lab\python codes\new2.py", line 2, in dijkstra
('D', 'C', 3)
Path: ('D', 'H', 2)
A -> D:
g = defaultdict(list)
NameError: global name 'defaultdict' is not defined
[Finished in 0.1s with exit code 1]
[shell_cmd: python -u "F:\ai lab\python codes\new2.py"]
[dir: F:\ai lab\python codes]
[path: C:\ProgramData\Oracle\Java\javapath;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS
Client\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program
Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files\Intel\Intel(R) Management Engine
Components\IPT;C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common Files\Intel\WirelessCommon\;C:\Program Files (x86)\MiKTeX
2.9\miktex\bin\;C:\Python27;C:\Users\Hp\AppData\Local\Microsoft\WindowsApps;C:\Users\Hp\AppData\Local\GitHubDesktop\bin]
Line 34, Column 23
Spaces: 4
Python
Type here to search
ENG
TN
9:05 AM
5/18/2018

4 Conclusion:

shortest path is calculated.