Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 7 REPORT

Ahmet Muzaffer DÜLGER
131044082

Course Assistant: Fatma Nur Esirci

1 Q1

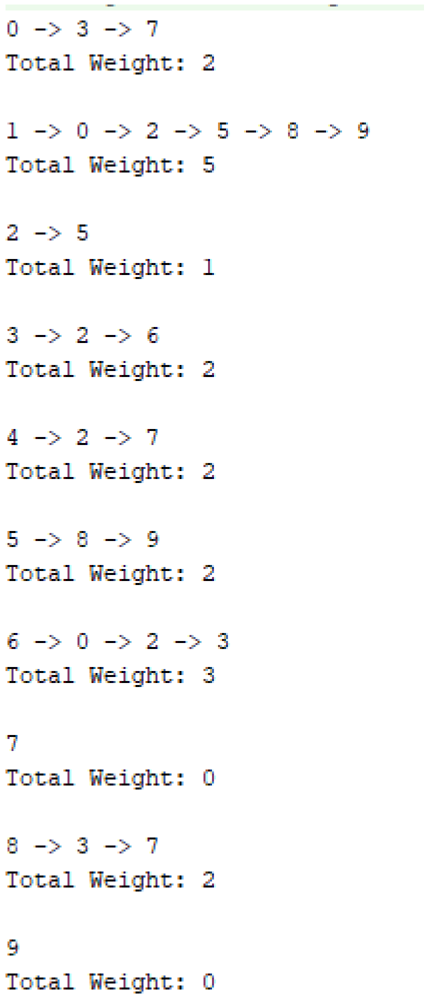
1.1 Problem Solution Approach

Explain simply graph creation, how to find shortest path.
Write roughly the changes you make.

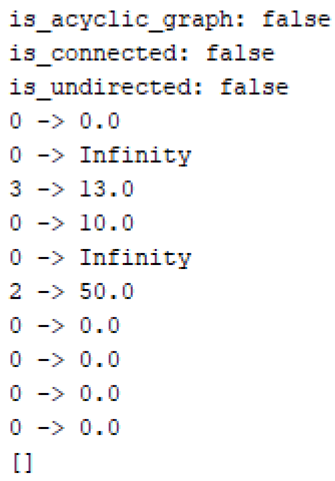
1.2 Test Cases

Show that this func results ->

- plot_graph
- is_undirected
- is_acyclic_graph
- shortest_path (use least 3 different label pair)



```
0 -> 3 -> 7
Total Weight: 2

1 -> 0 -> 2 -> 5 -> 8 -> 9
Total Weight: 5

2 -> 5
Total Weight: 1

3 -> 2 -> 6
Total Weight: 2

4 -> 2 -> 7
Total Weight: 2

5 -> 8 -> 9
Total Weight: 2

6 -> 0 -> 2 -> 3
Total Weight: 3

7
Total Weight: 0

8 -> 3 -> 7
Total Weight: 2

9
Total Weight: 0
```



```
is_acyclic_graph: false
is_connected: false
is_undirected: false
0 -> 0.0
0 -> Infinity
3 -> 13.0
0 -> 10.0
0 -> Infinity
2 -> 50.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
[]
```

2 Q2

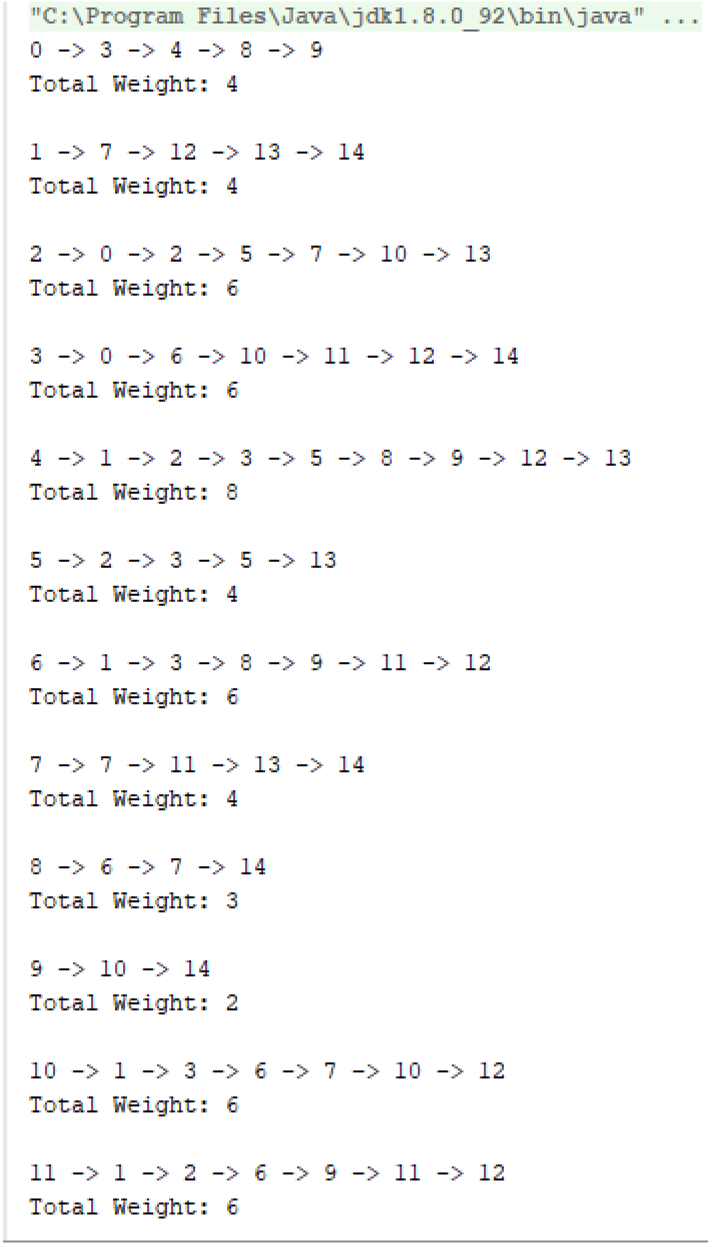
2.1 Problem Solution Approach

Explain simply graph creation, how to find shortest path.
Write roughly the changes you make.

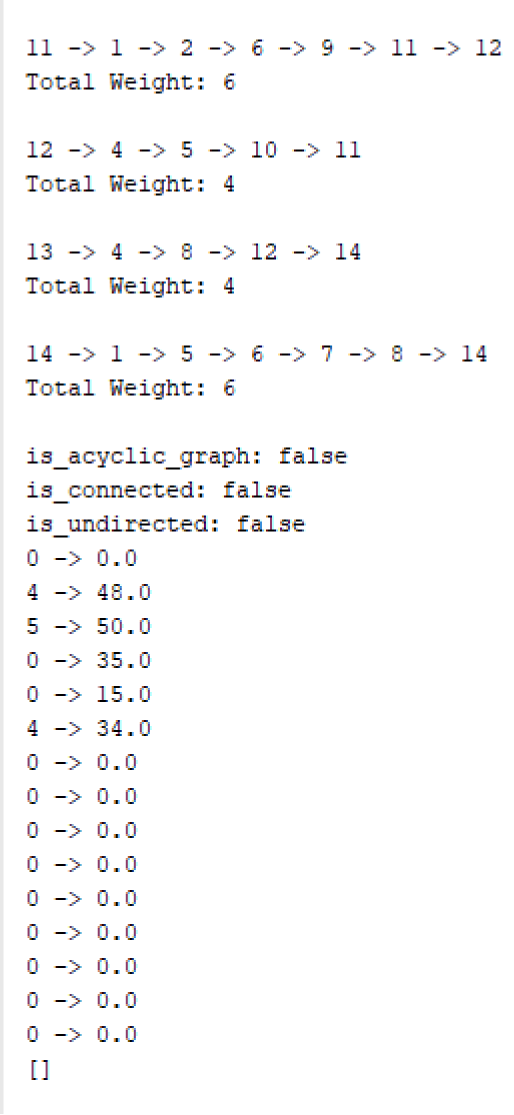
2.2 Test Cases

Show that this func results ->

- plot_graph
- is_undirected
- is_acyclic_graph
- is_connected function (use least 3 different label pair)



```
"C:\Program Files\Java\jdk1.8.0_92\bin\java" ...
0 -> 3 -> 4 -> 8 -> 9
Total Weight: 4

1 -> 7 -> 12 -> 13 -> 14
Total Weight: 4

2 -> 0 -> 2 -> 5 -> 7 -> 10 -> 13
Total Weight: 6

3 -> 0 -> 6 -> 10 -> 11 -> 12 -> 14
Total Weight: 6

4 -> 1 -> 2 -> 3 -> 5 -> 8 -> 9 -> 12 -> 13
Total Weight: 8

5 -> 2 -> 3 -> 5 -> 13
Total Weight: 4

6 -> 1 -> 3 -> 8 -> 9 -> 11 -> 12
Total Weight: 6

7 -> 7 -> 11 -> 13 -> 14
Total Weight: 4

8 -> 6 -> 7 -> 14
Total Weight: 3

9 -> 10 -> 14
Total Weight: 2

10 -> 1 -> 3 -> 6 -> 7 -> 10 -> 12
Total Weight: 6

11 -> 1 -> 2 -> 6 -> 9 -> 11 -> 12
Total Weight: 6
```



```
11 -> 1 -> 2 -> 6 -> 9 -> 11 -> 12
Total Weight: 6

12 -> 4 -> 5 -> 10 -> 11
Total Weight: 4

13 -> 4 -> 8 -> 12 -> 14
Total Weight: 4

14 -> 1 -> 5 -> 6 -> 7 -> 8 -> 14
Total Weight: 6

is_acyclic_graph: false
is_connected: false
is_undirected: false
0 -> 0.0
4 -> 48.0
5 -> 50.0
0 -> 35.0
0 -> 15.0
4 -> 34.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
0 -> 0.0
[]
```

3 Q3

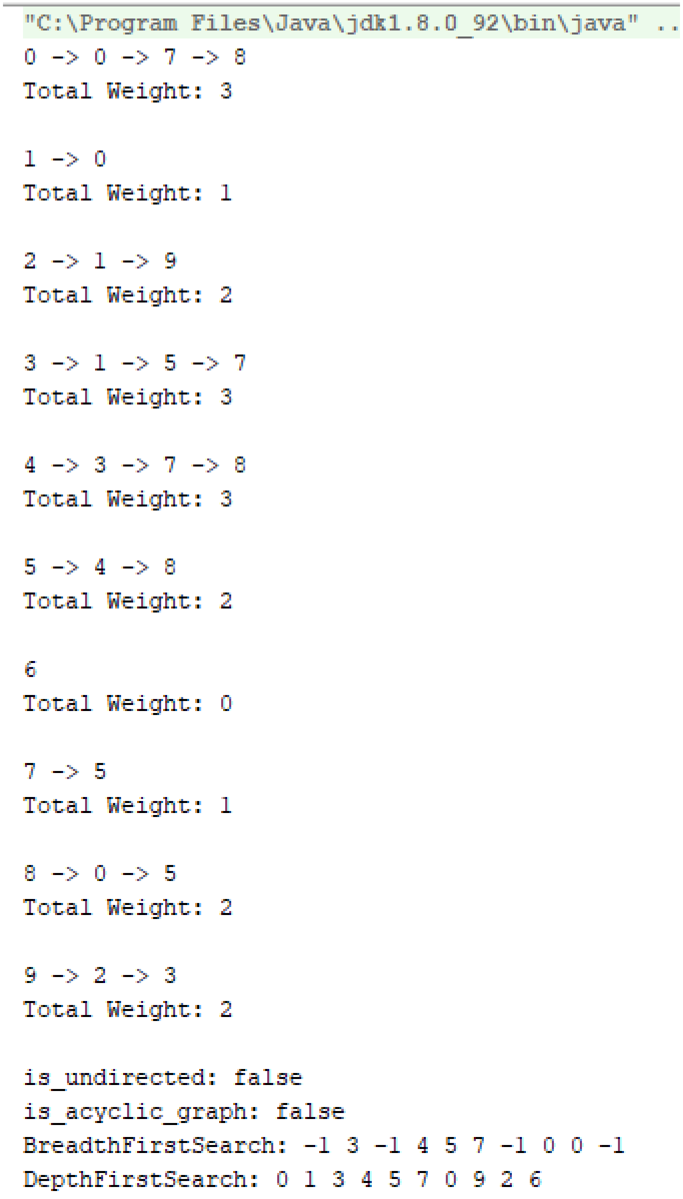
3.1 Problem Solution Approach

Explain simply graph creation, how to find shortest path.
Write roughly the changes you make.

3.2 Test Cases

Show that this func results ->

- plot_graph
- is_undirected
- is_acyclic_graph
- DepthFirstSearch (Show that spanning tree)
- BreathFirstSearch (Show that spanning tree)

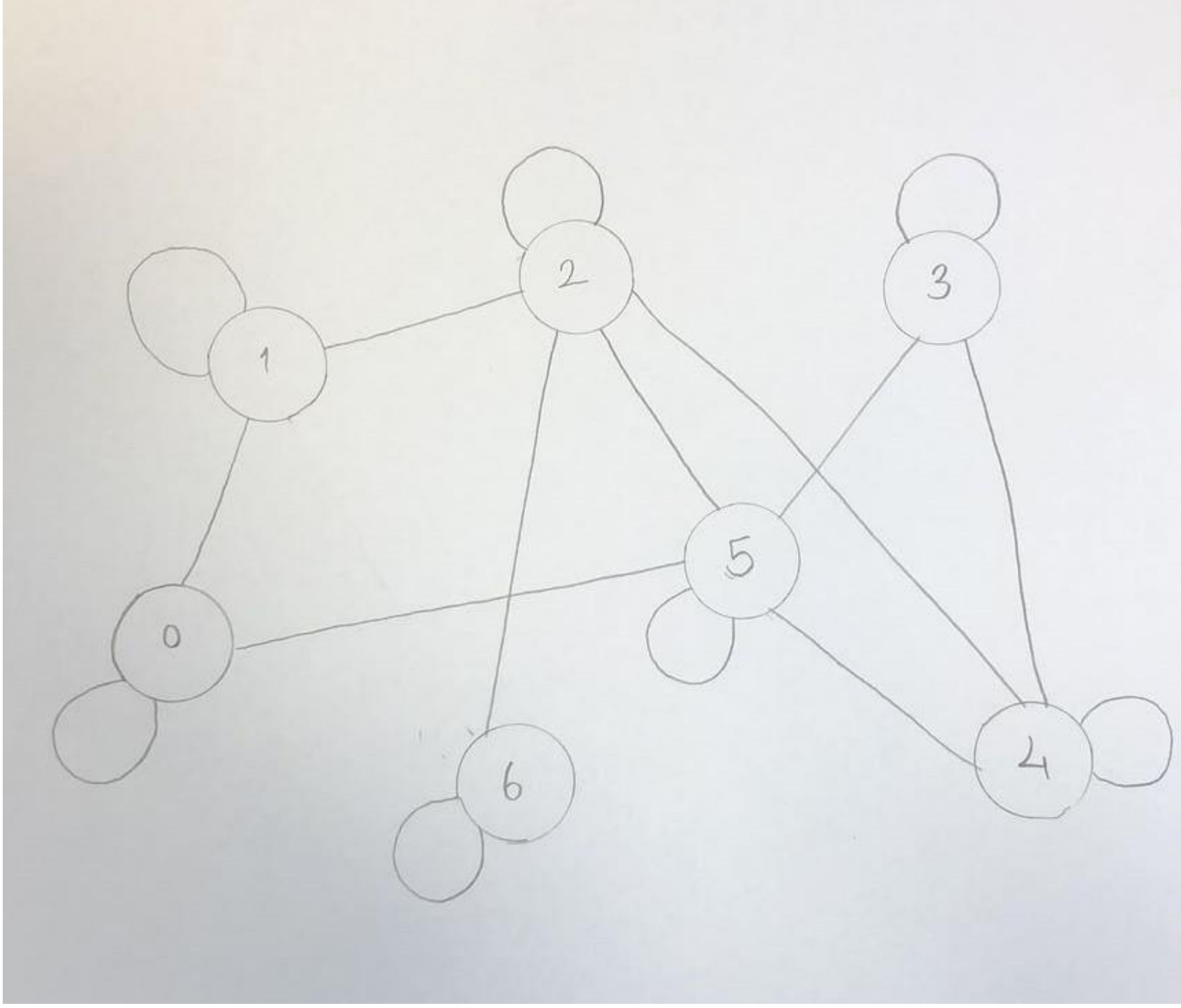


```
"C:\Program Files\Java\jdk1.8.0_92\bin\java" ..
0 -> 0 -> 7 -> 8
Total Weight: 3

1 -> 0
Total Weight: 1

2 -> 1 -> 9
Total Weight: 2

3 -> 1 -> 5 -> 7
Total Weight: 3

4 -> 3 -> 7 -> 8
Total Weight: 3

5 -> 4 -> 8
Total Weight: 2

6
Total Weight: 0

7 -> 5
Total Weight: 1

8 -> 0 -> 5
Total Weight: 2

9 -> 2 -> 3
Total Weight: 2

is_undirected: false
is_acyclic_graph: false
BreadthFirstSearch: -1 3 -1 4 5 7 -1 0 0 -1
DepthFirstSearch: 0 1 3 4 5 7 0 9 2 6
```

4 Q4

BFS ve DFS karşılaştırılması.

1) BFS'de ilk önce kök düğüm genişletilir, ardından kök düğümün tüm halefleri genişletilir ve sonraki adımda her düğümün tüm halefleri genişletilir, süreç hedefe ulaşana kadar devam eder. DFS'de kök düğümünü keşfeder ve hedefe ulaşılana kadar kök düğümden mümkün olduğunca uzaklaşırız.
2) BFS'de uzay karmaşıklığı, zaman karmaşıklığına kıyasla daha kritiktir. DFS'de daha az alan karmaşıklığı vardır, çünkü bir seferde kökten yaprak düğümüne sadece tek bir yolu depolaması gerekir.
3) BFS, FIFO(First In First Out) mantığı ile çalışırken DFS, LIFO(Last In First Out) mantığı ile çalışır.
4) BFS, DFS’den daha yavaş çalışır.
5) BFS, DFS'ye göre daha fazla bellek gerektirir.
6) BFS en kısa yolu bulmakta faydalıdır. DFS en kısa yolu bulmak için çok yararlı değildir.


2
3
1
5
0
6
4