

**Gebze Technical University  
Computer Engineering**

**CSE 222 - 2018 Spring**

**HOMEWORK 7 REPORT**

**Ramazan GUVENC  
161044037**

Course Assistant: Fatma Nur Esirci

# 1 Q1

This part about Question1 in HW7

## 1.1 Problem Solution Approach

Asagida resimlerde oluşturdüğüm graphin resimleri var. Kisaca grapha edge insert ederken once source u sonra destini yazarak ve istersek o yolun agirlık/zorluk vs. nasıl ifade ederseniz onuda belirterek ekliyoruz. Ben shortest path algorithm de kac br yol gitmesi gerektiğini bastırabildim fakat pathi bastıramadım.

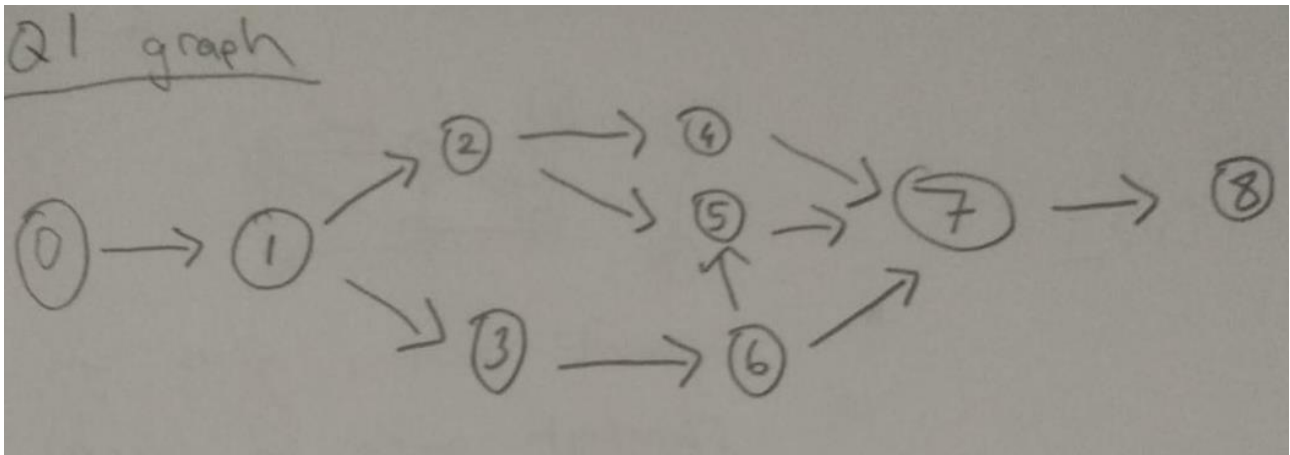
## 1.2 Test Cases

```
0 --->1
1 --->2
1 --->3
2 --->4
2 --->5
3 --->6
4 --->7
5 --->7
6 --->5
6 --->7
7 --->8
```

```
Is directed ? : true
Is acyclic ? : true
```

```
0 -- 6 16 br.
1 -- 4 13 br.
3 -- 7 17 br.
```

And this is my graph.



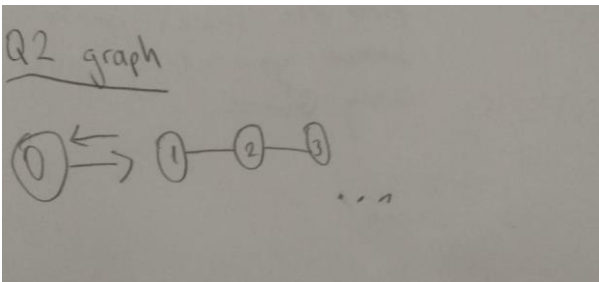
## 2 Q2

This part about Question2 in HW7

### 2.1 Problem Solution Approach

Matrix graph kullandigim için connected olup olmadiklarini anlamak baya kolay oldu. Diyelim  $x,y$  verticeslerinin bagli olup olmadigini öğrenmek istiyorsunuz.  $Matrix[x][y]$  ve  $matrix[y][x]$  ikiside 0 dan farkli değerlerse bu iki nokta baglidir.

### 2.2 Test Cases



Yukardaki graph bst tree worst case i gibi uc uca eklenmiş edgelerden oluşuyor.

```
Is directed ? : false
Is acyclic ? : true
Is 0 and 1 connected ? : true
Is 2 and 5 connected ? : false
Is 10 and 11 connected ? : true
```

Bunlarda test sonuclarim.

## 3 Q3

This part about Question3 in HW7

### 3.1 Problem Solution Approach

BFS ve DFS kodlarinin içine print koyup bastirdim. Spanning treeleri bastiramadim.

## 3.2 Test Cases

```
-----  
DFS yapiliyor  
0-->1  
1-->0  
1-->4  
4-->0  
4-->1  
4-->5  
5-->4  
4-->6  
6-->1  
6-->4  
6-->7  
7-->6  
7-->8  
8-->7  
8-->9  
9-->8  
1-->6  
0-->2  
2-->0  
2-->3  
3-->0  
3-->2  
0-->3  
0-->4  
  
-----  
Breadth First Search yapiliyor  
0-->1  
0-->2  
0-->3  
0-->4  
1-->0  
1-->4  
1-->6  
2-->0  
2-->3  
3-->0  
3-->2  
4-->0  
4-->1  
4-->5  
4-->6  
6-->1  
6-->4  
6-->7  
5-->4  
7-->6  
7-->8  
8-->7  
8-->9  
9-->8
```

## 4 Q4

BFS de root dan başlayıp her visit edebileceğin node u visit ede ede gider ve sonsuz döngüye girmek için daha önce gittiğin nodeları işaretlersin bir nevi ekmek kirintisi koyarsın maze de kaybolmamak ve sürekli aynı yerde dolaşmamak için.

DFS de ise rootdan olabildiğince uzaklaşmaya çalışırız.

DFS ile BFS arasındaki farklar şunlardır.

- 1- BFS de memory zamandan daha önemlidir. Mesela 7x7x7 zeka kupunde bfs yapamazsınız çünkü milyarlarca farklı durum var. Bu yüzden grapha bfs uygularken memoryye dikkat etmek zorundayız. DFS de ise bu memory problemi o kadar yoktur çünkü sadece leaf den node a kadar olan pathi hafızada saklarız. BFS de ise her yere yayıldığımız için neredeyse her pathi aynanda saklamak zorunda kalırız.
- 2- BFS queue kullanırken DFS stack kullanır(Quoradan yardım alınmıştır)
- 3- BFS daha yavaştır doğal olarak.
- 4- BFS shortest path bulunurken kullanılır çünkü tüm pathleri buluyorsunuz.