

Gebze Technical University  
Computer Engineering

CSE 222  
2017 Spring

HOMEWORK 09 REPORT

Elif Şeyma ARMAĞAN  
151044042

Course Assistant:  
Ahmet SOYYİĞİT

# Problem solutions approach

addRandomEdgesToGraph() metodunu yazarken öncelikle parametre olarak gelen edgeLimit ve 0 arasında bir random değer üretir. Daha sonra bu random sayı kadar dönen bir döngü içinde birbirinden farklı source ve destination lar üretilir. Ve eğer bu edge graph ın içinde yoksa eklenir. Sonuç olarak eklenebilen edge lerin sayısı return edilir.

breadthFirstSearch() metodunu yapmak için bir tane private breadth first search yapan bir metod yazdım. Bu private metod başlangıç vertex ini ve predecessor ların ekleneceği bir list alır ve return değeri graph ın bipartite olup olmadığını belirten boolean değişkendir. Bipartite kontrol etme işlemi de breadth first search ile yapılabildiğinden 2 işlem de bu metod aracılığıyla yapılmaktadır. Ders kitabından aldığım breadth first search kodunu graph içerisinde çember(cycle) olup olmadığını kontrol edecek şekilde değiştirdim. Eğer içinde cycle varsa bu graph bipartite değildir. Public olan breadthFirstSearch() metodu private olan metodu çağırır ve predecessor list ini array e çevirip return eder.

getConnectedComponentUndirectedGraph() metodu graph içerisinde birbirine bağlı olan alt graphları bulur. Bu işi yapmak için depth first search kullandım. Yazdığım dfsRec() metodu başlangıç vertex i, vertexlerin ziyaret edilip edilmediğini tutan boolean array i ve ziyaret edilen node ları birbirine bağlayan edge lerin tutulacağı list i parametre olarak alır ve klasik depth first search uygular.

Her bir ziyaret edilmemiş vertex için dfsRec() metodu çağırılır ve o vertexin bağlı olduğu tüm edge leri elde ederiz. Bu edge lerdeki unique vertexleri(source ve destination) alırız. Bu vertexleri 0 dan başlayacak şekilde sayılara map ederiz. Böylelikle graph daki vertex sayısını gereksiz fazla yapmamış oluruz.

Daha sonra elde ettiğimiz vertexler kadar yeni bir graph oluştururuz. Yeni oluşturulan graph ın list mi yoksa matrix mi olacağına instance of kullanarak karar verdim. Son olarak dfsRec() metodundan gelen edge leri map i kullanarak yeni graph a ekledim.

isBipartiteUndirectedGraph() metodu, raporumdaki breadth first search implementasyonu kısmında anlatılan private breadth first search metodunun sonucunu return eder. Bu metodun nasıl çalıştığı ise yine raporumdaki breadth first search implementasyonu kısmında anlatılmıştır.

writeGraphToFile() metodunu yazarken öncelikle AbstractGraphExtended class ında toString metodunu override ettim. toString metodu ödevde istenen formatta bir string oluşturup return eder. Dolayısıyla writeGraphToFile() metodu içerisinde tek yaptığım toString den gelen string i dosyaya yazmak oldu.

# Test Cases

Ödev pdf inde istendiği şekilde her metod 2 farklı dosya için ve list ve matrix için test edilmiştir. Testlerin sonuçları 1 ve 3. Metod için dosyaya, 2 ve 4 için ise ekrana basılmıştır. 1 ve 3 ün oluşturduğu dosyaların isimleri de ekrana basılmıştır.

getConnectedComponentUndirectedGraph() metodunu test ederken dosyadan okuduğu graph içinde bulunduğu her bir alt graph ı ayrı bir dosyaya yazmaktadır. O yüzden bu method test edilirken alt graph sayısı kadar dosya oluşmaktadır.

## Running command and results

### addRandomEdgesToGraph() için test sonuçları

#### Test dosyaları:

test.txt		test2.txt	
MatrixGraph.java ×	Main.java ×	Graph.java ×	Main.java ×
test.txt ×		test.txt ×	test2.txt ×
1	9	1	10
2	0 1	2	0 1
3	0 3	3	1 2
4	1 2	4	2 3
5	1 4	5	1 0
6	1 5	6	7 0
7	2 5	7	0 7
8	3 6	8	2 8
9	4 6	9	3 4
10	4 7	10	4 6
11	5 7	11	5 7
12	6 8	12	5 9
13	7 8	13	8 9

#### Terminal çıktısı:

```
Testing addRandomEdgesToGraph() with 6 for List Graph from test.txt (for directed graph)
Result is saved in resultListForAdd.txt
Testing addRandomEdgesToGraph() with 6 for List Graph from test2.txt (for undirected graph)
Result is saved in resultListForAdd2.txt

Testing addRandomEdgesToGraph() with 5 for Matrix Graph from test.txt (for directed graph)
Result is saved in resultMatrixForAdd.txt
Testing addRandomEdgesToGraph() with 5 for Matrix Graph from test2.txt (for undirected graph)
Result is saved in resultMatrixForAdd2.txt
```

## Output dosya içerikleri :

resultListForAdd.txt

1	9
2	0 1
3	0 3
4	0 6
5	1 2
6	1 4
7	1 5
8	2 5
9	3 6
10	3 4
11	4 6
12	4 7
13	5 7
14	6 8
15	7 8
16	7 2
17	8 4
18	

resultListForAdd2.txt

1	10
2	0 1
3	0 1
4	0 7
5	0 7
6	1 0
7	1 2
8	1 0
9	1 7
10	2 1
11	2 3
12	2 8
13	3 2
14	3 4
15	4 3
16	4 6
17	5 7
18	5 9
19	6 4
20	7 0
21	7 0
22	7 5
23	7 1
24	8 2
25	8 9
26	9 5
27	9 8
28	

resultMatrixForAdd.txt

1	9
2	0 1
3	0 3
4	0 5
5	1 2
6	1 4
7	1 5
8	2 5
9	3 6
10	4 6
11	4 7
12	5 7
13	6 8
14	7 8
15	

resultMatrixForAdd2.txt

1	10
2	0 1
3	0 3
4	0 7
5	1 0
6	1 2
7	2 1
8	2 3
9	2 8
10	3 0
11	3 2
12	3 4
13	4 3
14	4 6
15	5 7
16	5 9
17	6 4
18	7 0
19	7 5
20	8 2
21	8 9
22	9 5
23	9 8
24	

## breadthFirstSearch() için test sonuçları

### Test dosyaları:

test.txt

MatrixGraph.java	Main.java	test.txt
1	9	
2	0 1	
3	0 3	
4	1 2	
5	1 4	
6	1 5	
7	2 5	
8	3 6	
9	4 6	
10	4 7	
11	5 7	
12	6 8	
13	7 8	

bipartiteTest.txt

test2.txt	bipartiteTest.txt	testConnected.txt
1	11	
2	1 4	
3	2 4	
4	3 4	
5	4 5	
6	5 6	
7	5 7	
8	5 8	
9	5 9	
10	8 10	
11	9 10	

### Terminal çıktısı:

```
Testing breadthFirstSearch() with 0 for List Graph from test.txt (for directed graph)
Result: [-1, 0, 1, 0, 1, 1, 3, 4, 6]
Testing breadthFirstSearch() with 0 for List Graph from test2.txt (for undirected graph)
Result: [-1, 0, 1, 2, 3, 7, 4, 0, 2, 5]

Testing breadthFirstSearch() with 0 for Matrix Graph from test.txt (for directed graph)
Result: [-1, 0, 1, 0, 1, 1, 3, 4, 6]
Testing breadthFirstSearch() with 0 for Matrix Graph from test2.txt (for undirected graph)
Result: [-1, 0, 1, 2, 3, 7, 4, 0, 2, 5]
```

## getConnectedComponentUndirectedGraph() için test sonuçları

### Test dosyaları:

testConnected.txt

test2.txt	bipartiteTest.txt	testConnected.txt
1	7	
2	0 1	
3	0 2	
4	1 2	
5	3 4	
6	5 6	

### Terminal çıktısı:

```
Testing getConnectedComponentUndirectedGraph() for List Graph from testConnected.txt(for undirected graph)
Results are saved to connectedResListUndirected[num] files.
Testing getConnectedComponentUndirectedGraph() for Matrix Graph from testConnected.txt (for undirected graph)
Results are saved to connectedResMatrixUndirected[num] files.
```

### Output dosya içerikleri :

List Graph için sonuçlar

Burada sadece 0, 1 gibi olmasının nedeni elemanları yeni oluşan graph ta 0 dan başlayarak map etmemdir.

C:\Users\elif\IdeaProj...	C:\Users\elif\Idea...	C:\Users\elif\IdeaPr...
File Edit Selection Find View Goto Tools Project Preferences Help	File Edit Selection Find View Goto Tools Project Preferences Help	File Edit Selection Find View Goto Tools Project Preferences Help
connectedResListUndirected1.txt	connectedResListUndirected2.txt	connectedResListUndirected3.txt
1 3	1 2	1 2
2 0 1	2 0 1	2 0 1
3 1 0	3 1 0	3 1 0
4 1 2	4	4
5 2 1		
6		
Line 4, Column 4	Line 4, Column 1	Line 4, Column 1
Tab Size: 4	Tab Size: 4	Tab Size: 4

## Matrix Graph için sonuçlar

Burada sadece 0, 1 gibi olmasının nedeni elemanları yeni oluşan graph ta 0 dan başlayarak map etmemdir.

```
connectedResMatrixUndirected1.txt
1 3
2 0 1
3 1 0
4 1 2
5 2 1
6

nnectedResMatrixUndirected2.txt
1 2
2 0 1
3 1 0
4

connectedResMatrixUndirected3.txt
1 2
2 0 1
3 1 0
4 |
```

## isBipartiteUndirectedGraph() için test sonuçları

### Test dosyaları:

bipartiteTest.txt

test2.txt	bipartiteTest.txt	testConnected.txt
1	11	
2	1 4	
3	2 4	
4	3 4	
5	4 5	
6	5 6	
7	5 7	
8	5 8	
9	5 9	
10	8 10	
11	9 10	

### Terminal çıktısı:

```
Testing isBipartiteUndirectedGraph() for List Graph from bipartiteTest.txt
Result: true
Testing isBipartiteUndirectedGraph() for List Graph from test2.txt
Result: false

Testing isBipartiteUndirectedGraph() for Matrix Graph from bipartiteTest.txt
Result: true
Testing isBipartiteUndirectedGraph() for Matrix Graph from test2.txt
Result: false
```