

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 09 REPORT

Büşra ARSLAN
131044021

Course Assistant: Ahmet SOYYİĞİT

TABLE OF CONTENTS

1)Problem Solutions Approach.....

2)Test Cases.....

3) Running command and results.....

Not: JavaDoc projenin içerisinde Javadoc klasörünün içerisinde mevcuttur. Github' a tüm partlar upload edilmiştir.

1. Problem Solutions Approach

İlk başta bir input file den graph oluşturmak için bir edge'in source ve destination vertexleri ve total vertex sayısı okunur.

- **public int addRandomEdgesToGraph (int edgeLimit);**

Bu fonksiyonda 0 ile edgeLimit arasında random bir sayı kadar edge eklenir.

Ben random eklemeyi şu şekilde yaptım:

Directed olarak düşünerek iki for döngüsünün içinde tek tek tüm vertexler arasında dge var mı diye kontrol ettim eğer varsa devam ettim bakmaya döngü içerisinde ama yoksa edge oraya edge ekledim . Bu edge ekleme işlemi edgeLimit-1 kadar oldu random'ımı bu şekilde belirlemek mantıklı geldi. Eklenen edge sayısını return edip dosyaya bastırdım.

Dosya formatım: `graph_addRandomEdgeGraph_out_m1_1.txt`

- **public int [] breadthFirstSearch (int start);**

Burda kitabın kodunu kullandım. Static'i kaldırdım ve Parametre olarak aldığı graphı sildim. Main üzerinde belirlenen graph üzerinden çalışır hale getirdim. Vertexlerin parentlarını bir array olarak return ettirip console'a bastırdım.

- **public Graph [] getConnectedComponentUndirectedGraph ();**

Burda graph arrayi tutulcak ve bu return edilecektir. Bulduğum her farklı renkli graphı bu graph arrayine eklidim. Ve her bir graphı başka dosyaya bastırdım. Oluşturulacak bir graph içinde connected şekildedir. Connected olamdığı an başka bir graph oluşturulacak şeklinde algoritmamı düzenledim.

Not: Ama graphların başında ilk baştaki graphların vertex sayısını bastım.

Not: Graphları undirected olarak düşündüğümüz için mesela 0 -1 vertexleri bir graph bunu bulduğunda hem 0-1 hem 1-0 dosyaya bastırdım. Çünkü undirected graph olarak düşündüğüm için bu şekilde basıldı.

m kaç tane graph varsa o kadar dosya oluşturulacağı için bir for döngüsü içinde artırılıyor.

Dosya formatı: `graph_getConnectedComponent_out_graph" + m + ".txt"`

- **public boolean isBipartiteUndirectedGraph ();**

Bu fonksiyonda elimizdeki graphı 2 ayrı renge boyararak 2 graph elde ediyoruz. Oluşturulacak graphlar şu şekilde oluşacaktır:

Aralarında edge olmamak aynı renkteki graphların. Karşı renkteki graphlarla eşleşecekler. Connected halinde de olabilir isBipartite graphlar.

Renk çakışması olduğu an isBipartite değildir denir.

Eğer graphımız isBipartite ise true değil ise false döndürülerek console a basılmıştır. İki ayrı input test dosyası ile test edilmiştir. Console da çıktıları mevcuttur.

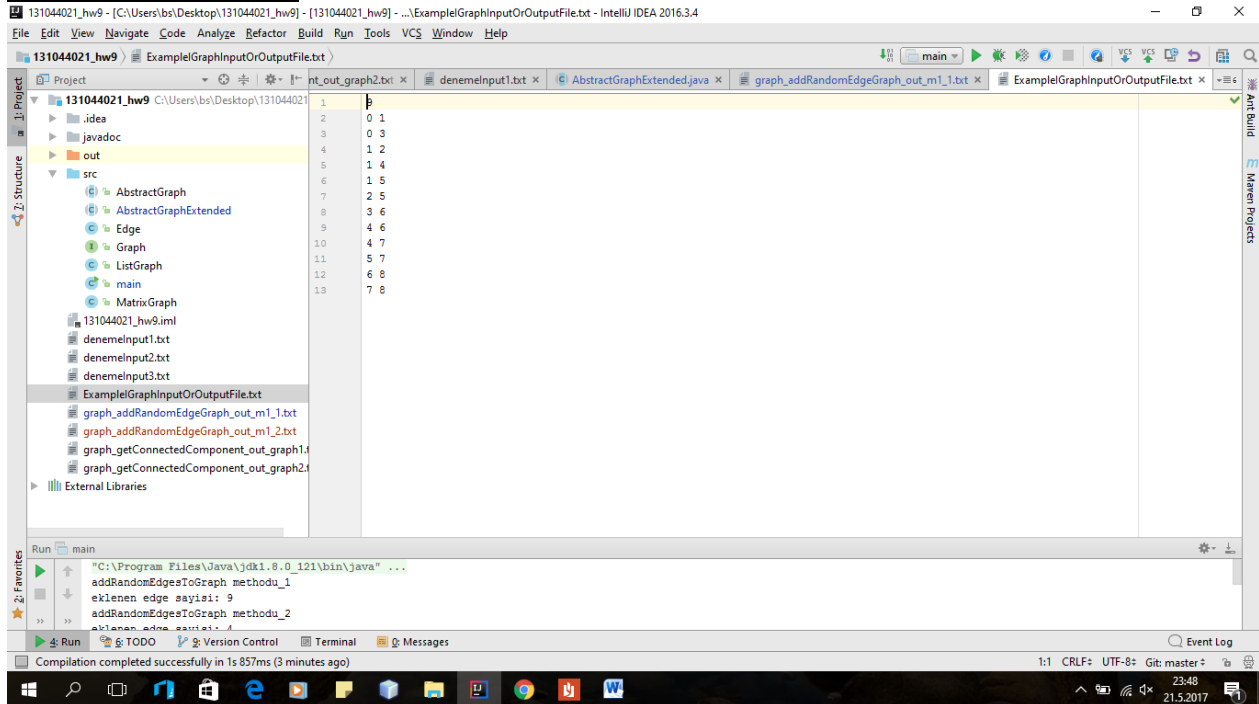
Not: <http://www.geeksforgeeks.org/bipartite-graph/> burdan yardım alınmıştır.

- **public void writeGraphToFile (String fileName);**

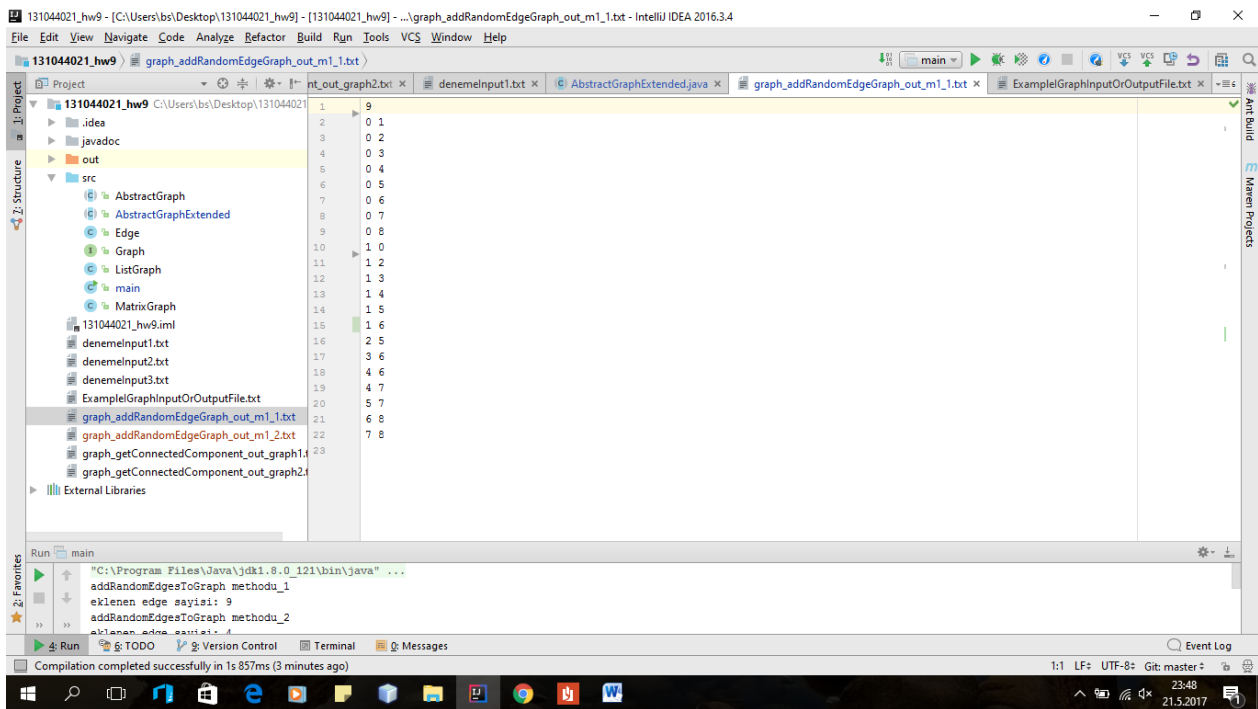
Bu methodu kullanmadaki amacımız dosyaya yazılacak fonksiyonları test etmektir. isEdge fonksiyonu ile vertexlerin arasında edge olanlar kontrol edilip eğer edge varsa bu vertex değerleri ve dosyanın en başına toplam vertex sayısı yazılmaktadır.

2)Test Cases

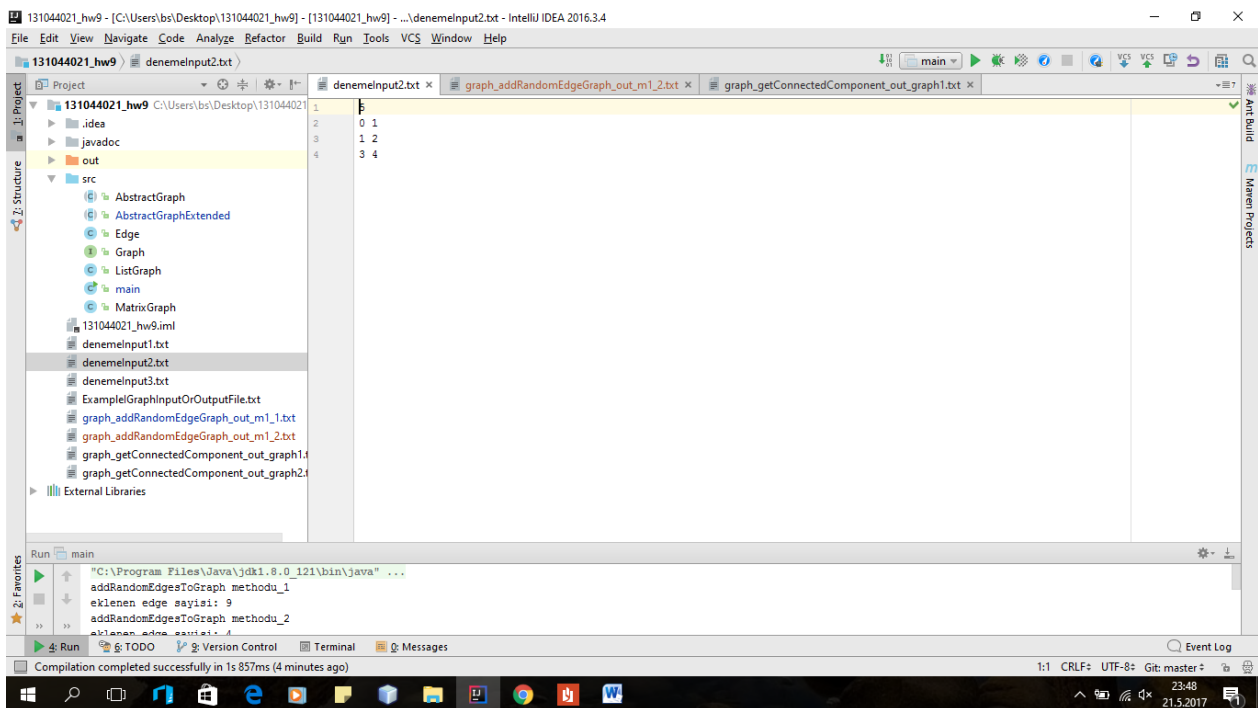
addRandom input



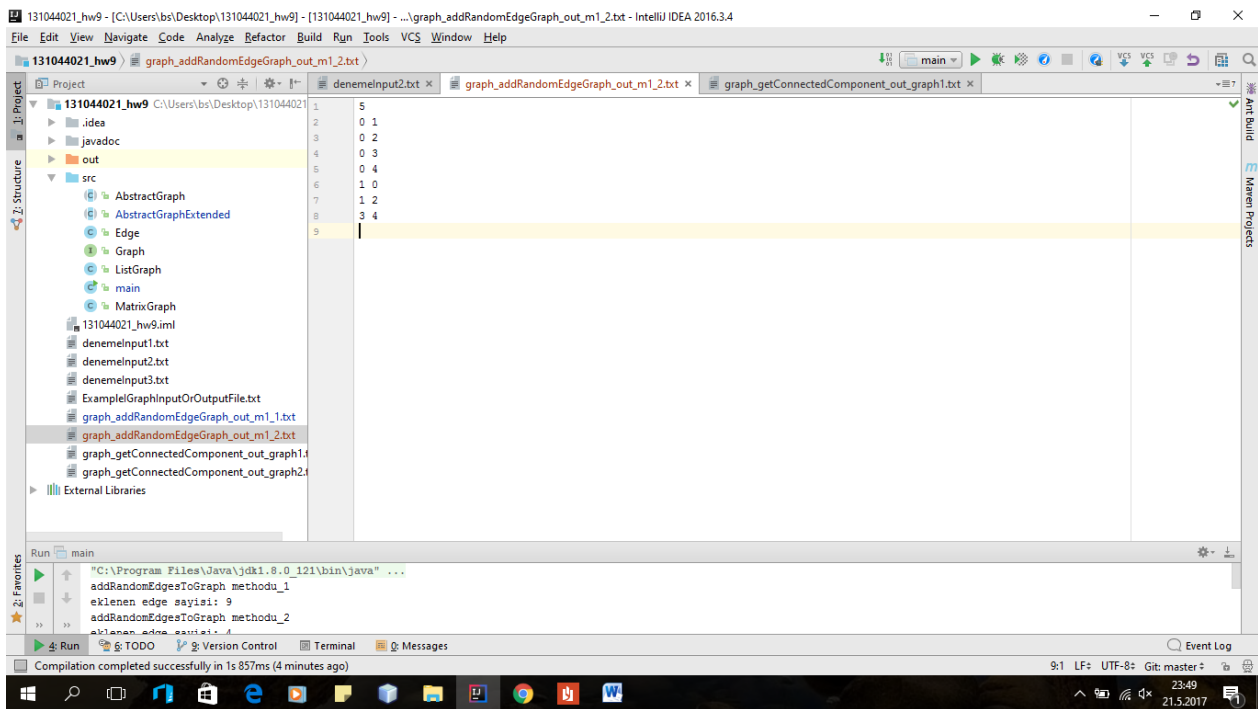
Output



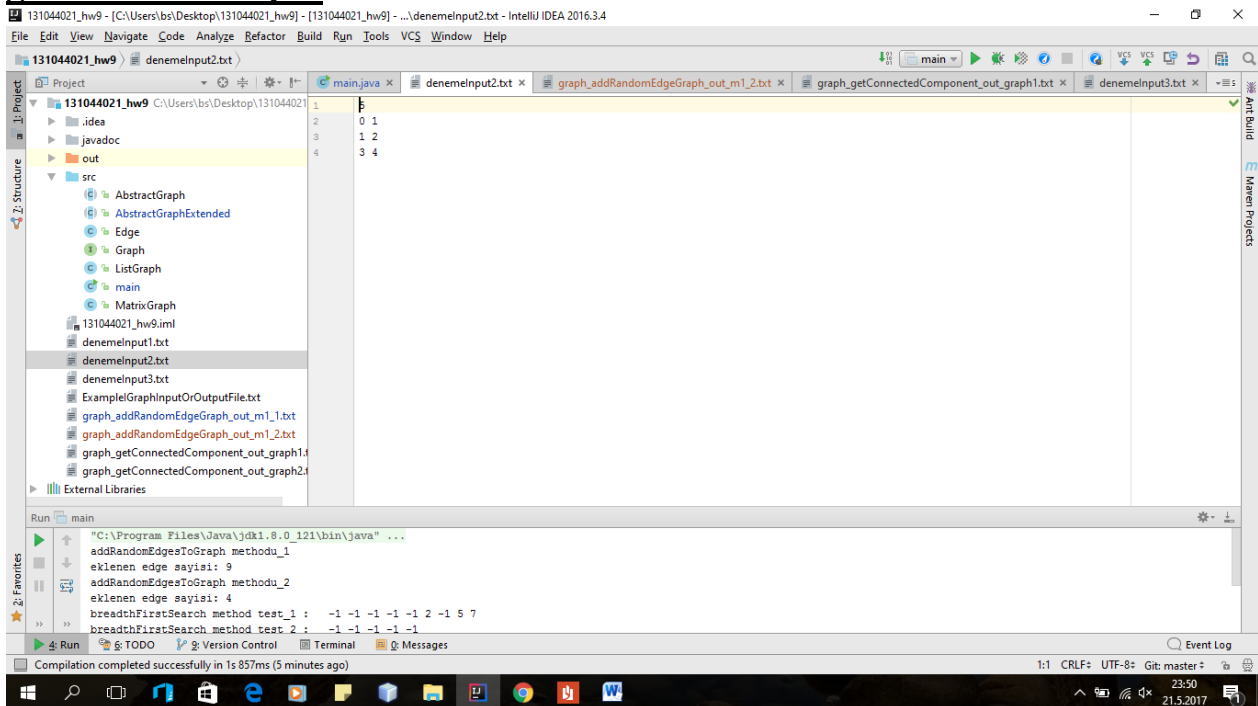
addRandom input t 2



Outputu

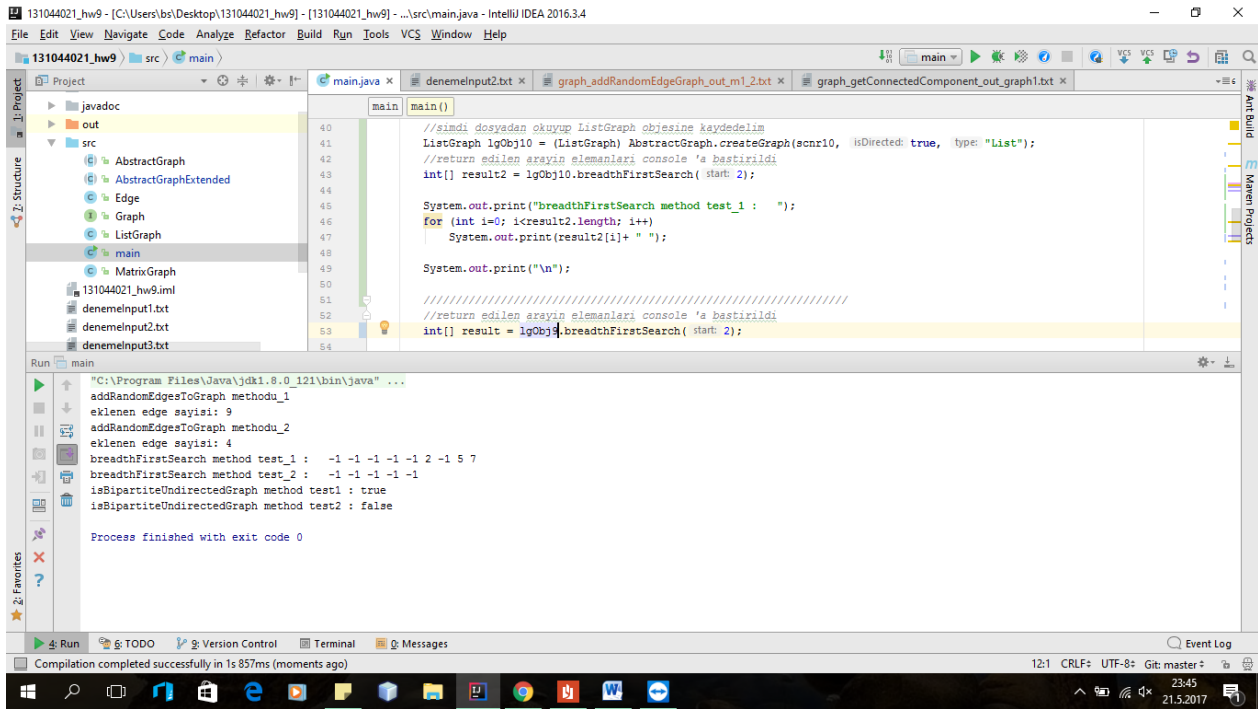


getCommenet input



3) Running command and results

Console çıktıları



```
131044021_hw9 - [C:\Users\bs\Desktop\131044021_hw9] - [131044021_hw9] - ...src\main.java - IntelliJ IDEA 2016.3.4
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

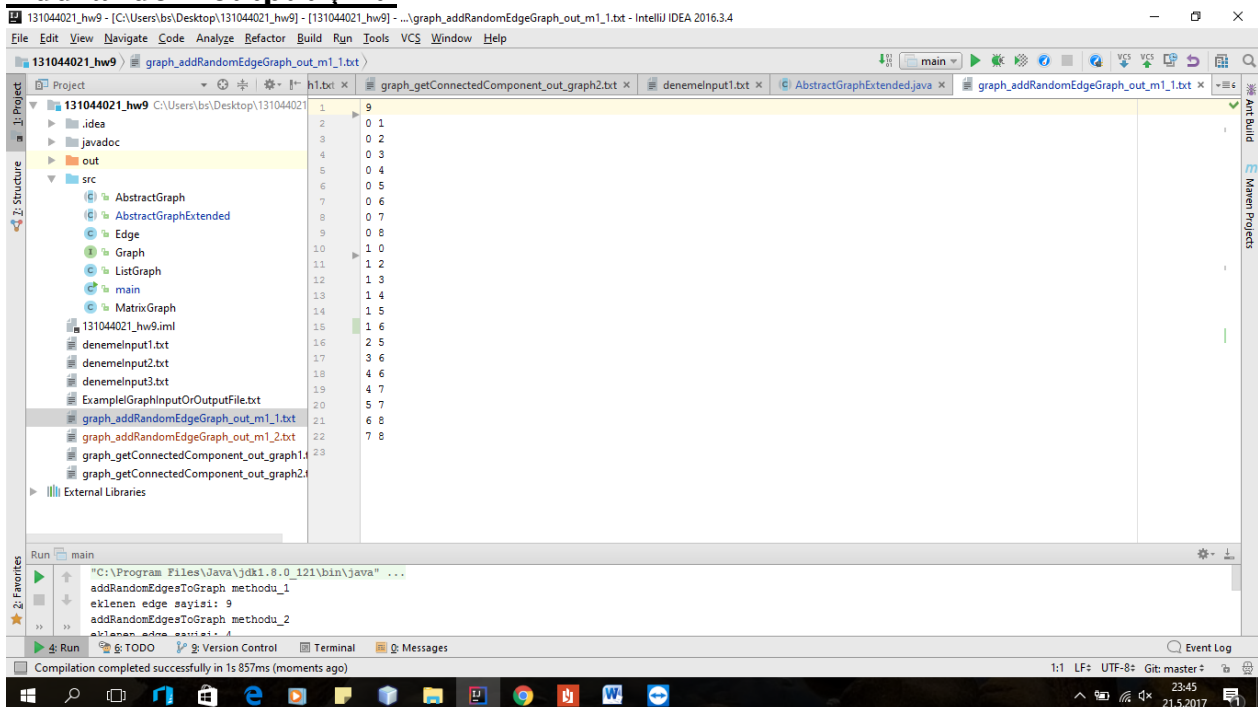
Project Structure:
- javadoc
- out
- src
  - AbstractGraph
  - AbstractGraphExtended
  - Edge
  - Graph
  - ListGraph
  - main
  - MatrixGraph
- 131044021_hw9.iml
- denemeInput1.txt
- denemeInput2.txt
- denemeInput3.txt

main.java:
40 //simdi dosyadan okuyup ListGraph objesine kaydedelim
41 ListGraph lqObj10 = (ListGraph) AbstractGraph.createGraph(scnr10, isDirected: true, type: "List");
42 //return edilen arrayin elemanlari console 'a bastirildi
43 int[] result2 = lqObj10.breadthFirstSearch( start: 2);
44
45 System.out.print("breadthFirstSearch method test_1 : ");
46 for (int i=0; i<result2.length; i++)
47     System.out.print(result2[i]+ " ");
48
49 System.out.print("\n");
50
51 //return edilen arrayin elemanlari console 'a bastirildi
52 int[] result = lqObj10.breadthFirstSearch( start: 2);
53
54

Run console:
"C:\Program Files\Java\jdk1.8.0_121\bin\java" ...
addRandomEdgesToGraph methodu_1
eklenen edge sayisi: 9
addRandomEdgesToGraph methodu_2
eklenen edge sayisi: 4
breadthFirstSearch method test_1 : -1 -1 -1 -1 -1 2 -1 5 7
breadthFirstSearch method test_2 : -1 -1 -1 -1 -1
isBipartiteUndirectedGraph method test1 : true
isBipartiteUndirectedGraph method test2 : false

Process finished with exit code 0
```

Addrandom output çıktı



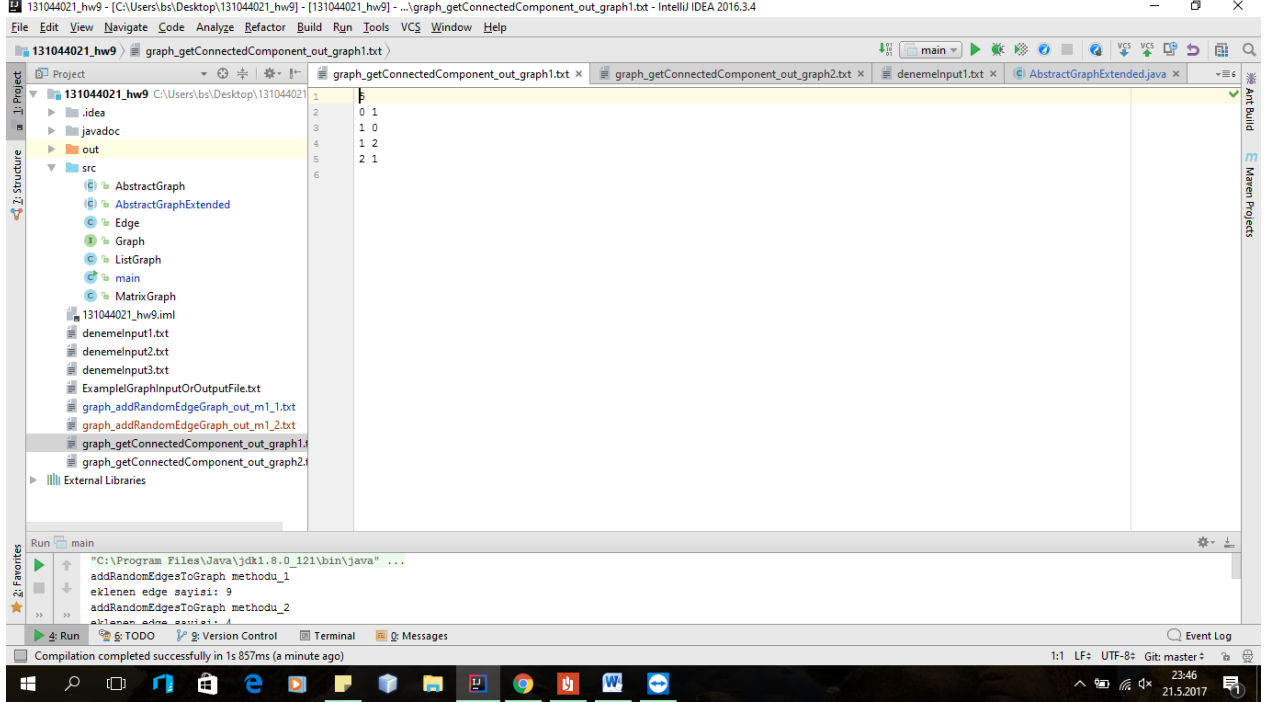
```
131044021_hw9 - [C:\Users\bs\Desktop\131044021_hw9] - [131044021_hw9] - ...graph_addRandomEdgeGraph_out_m1_1.txt - IntelliJ IDEA 2016.3.4
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Project Structure:
- .idea
- javadoc
- out
- src
  - AbstractGraph
  - AbstractGraphExtended
  - Edge
  - Graph
  - ListGraph
  - main
  - MatrixGraph
- 131044021_hw9.iml
- denemeInput1.txt
- denemeInput2.txt
- denemeInput3.txt
- ExampleGraphInputOrOutputFile.txt
- graph_addRandomEdgeGraph_out_m1_1.txt
- graph_addRandomEdgeGraph_out_m1_2.txt
- graph_getConnectedComponent_out_graph1.txt
- graph_getConnectedComponent_out_graph2.txt
- External Libraries

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

Run console:
"C:\Program Files\Java\jdk1.8.0_121\bin\java" ...
addRandomEdgesToGraph methodu_1
eklenen edge sayisi: 9
addRandomEdgesToGraph methodu_2
eklenen edge sayisi: 4
```

getConnectedComponent çıktı 2 graph oluşmuştu 1. si



getConnectedComponent çıktı 2 graph oluşmuştu 2. Si

