

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 9 REPORT

FÜRKAN YILDIZ
141044031

Course Assistant: Ahmet SOYYİĞİT

1. Problem Solutions Approach

addRandomEdgesToGraph

0 ile parametrede verilen sayı arasında random integer değeri üretilir, üretilen değeri eğer vertex sayısından büyük eşitse yada 0 ise maksimum 100 kere daha bu random sayı yenilendi, 100 denemenin ardından aynı durum devam ediyor ise exception fırlatıldı (0 olursa ekleme yapılamayacak, vertex sayısından büyük yada eşit olursa yine exception fırlatılacak) doğru aralıkta random numara üretildikten sonra ise, source ve destination değerleride yine aynı şekilde random olarak üretilecek, burada dikkat edilecek olan şeyler source nin yada destinationun vertex sayısına eşit yada büyük olmaması ve birbirlerine eşit olmamalarıdır. Eğer böyle bir durum gerçekleşirse 100 iterasyona kadar yeniden random sayı üretiliyor, 100 iterasyondan sonra hala aynı durum devam ediyor ise `IllegalArgumentException` exceptioni fırlatılıyor.

breadthFirstSearch

Kitaptaki algoritmadan yararlanıldı. Her vertex için bir identified arrayi ve birde parent arrayi tutuldu. Identified arrayi gezilen vertexlerin indexini true yaparken parent arrayine de gezilen vertexlerin parent vertexleri koyuldu. Parent vertexleri koyulurken bread first search algoritmasına göre önce verilen vertex, sonra o vertexin komşuları daha sonra o komşuların komşuları gezildi. Bu işlem queue ile yapıldı. Her vertex tek tek queue ye koyuldu ve bir vertex queue den çıkartılırken onun komşu vertexleri queue ye eklendi ve eklenirken onun komşusu olan parenti parent arrayine yazıldı. Bu şekilde tüm graph dolaşarak parent arrayi oluşturuldu ve return edildi.

getConnectedComponentUndirectedGraph

Öncelikle tüm graphı dolaşarak tüm edgeleri içeren bir edge arrayi oluşturuldu. Bu edge arrayinde source büyük, destination küçük seçildi ve source ve destination değerlerinin inverseleri (destination, source) eklenmedi.

Tüm edgeleri içeren arrayden sourcesi 0 olan edgeler çıkartılarak queue'ye koyuldu ve ardından bu edge çıkartılırken edgenin komşu edgeleri queue'ye eklendi. Queue'ye eklenen edge tüm edgeleri tuttuğumuz arrayden çıkartıldı ve graph arrayimize eklendi bu işlem'e queue boşalana kadar devam edildi. Queue boşaldıktan sonra ise sourcesi 1 olan edgeler için aynı işlemler yapıldı ta ki vertex sayısına kadar, graph'a ekleme yapılırken ise queue'den gelen komşular aynı graph arrayine eklenirken, sonradan gelen ve önceki graphlar ile ortak elemanı bulunmayanlar yeni graph arrayi oluşturularak ona eklendi.

Örneğin edge arrayimiz (0,1) (1,2) (3,4) ise (0,1) edge arrayinden çıkartılarak queue'ye eklendi, daha sonra (0,1) çıkartılırken 1'i içerdiğinden dolayı (1,2) arrayden çıkartılarak queue'ye ekledi ve queueden çıkan (0,1) ilk graphımızın elemanı oldu. (1,2) çıkartılırken ise (0,1) e komşu olduğu ve herhangi başka komşu içermediği için aynı graph'a eleman olarak eklendi ve queue'miz boşaldı. Queue boşaldıktan sonra döngünün en başına gidildi ve döngü 1 numaralı vertex'i içerek edge var mı diye edge arrayimizi gezdi bir şey bulamadı geri döndü, ardından 2 için aynı şey oldu ve 3'e sıra geldiğinde (3,4) queue'ye eklendi komşu edgesi olmadığından dolayı direkt çıkartıldı ve graph arrayinde bu edgenin source veya desti olup olmadığına bakıldı, bulunmadığı için yeni bir graph arrayi oluşturuldu (bu bir flag tutularak sağlandı) ve (3,4) edgesi yeni graph arrayimize atıldı böylece (0,1) (1,2) ve (3,4) olmak üzere 2 adet *connected component* graph elde ettik.

Bu işlemlerin ardından elde ettiğimiz graphları tek tek "findMinArr" methoduna yolluyoruz bu method bize, graphların içerdikleri vertexleri sıralı bir şekilde array olarak dönderiyor. Bu arrayi kullanarak ilk oluşturduğumuz graph arrayi gibi yeni bir graph arrayi oluşturuyoruz ve bu yeni oluşturduğumuz graph arrayine önceki graph arrayindeki elemanları tek tek atıyoruz. Bu işlem sırasında ise sıralı arrayimizi kullanıyoruz, yani

örnek vermek gerekirse ilk graphımız (0,2)(1,3) ikinci graphımız (4,5) (5,7)(6,7) olsun biz graphları tek tek "findMinArr" methoduna gönderiyoruz ve örneğin ikinci graph için (4,5,6,7) arrayi dönderiyor bu arrayi kullanarak yeni graph arrayimizin ikinci graphına (4,5) edgesini eklerken 4'ün ve 5'in minimum arraydeki indexlerine göre yeni graphımıza ekliyeceğiz yani (0,1)(1,3)(2,3) şeklinde ve weightlerini ise aynı vereceğiz böylece graphların bağlantıları ve ilişkileri değişmeden graphları bölmüş olduk.

isBipartiteUndirectedGraph

Graphın bipartite olup olmadığı kontrol edilmek için graph daki her bağımsız bileşenlere tek tek bakılmalı, bunun içinde "getConnectedComponentUndirectedGraph" methodundan yararlanıldı. Tek tek tüm bağımsız graphlar için işlem yapıldı, yapılan işlem ise: Graphların vertex sayısı kadar renk arrayi(integer bir array, integer değerleri bir renk olarak düşünüldü) tutmak, bu renk arraylerine ilk değer olarak -1 atamak ve ardından her gezilen vertex için o renk değerini değiştirmek, eğer komşu 2 vertex aynı renkte ise graph bipartite değildir, böyle bir koşul yoksa bipartitedir.

Graph gezilmeli bu işlemleri gerçekleştirebilmek için, 0. vertexden başlandı gezilmeye 0 bir queue'ye atıldı ve renk arrayindeki 0. indexe de 1 atandı. Daha sonra queue boş olana kadar yani diğer bir değişle tüm graph gezilene kadar iterator yardımıyla graph üzerinde gezildi. (her yeni vertex queue'ye atılarak) Eğer gezilen vertexin (source) komşusu (dest) nun color arrayindeki değeri -1 ise o değer 1 - destination'un color arrayindeki değeri ile değiştirildi ve ardından destination ve source indexlerinin color arraylerindeki değerlerinin aynı olup olmadığına bakıldı, eğer aynı iseler graph bipartite özelliğine sahip olmayacaktır. Bipartite ise graph komşu vertexlerin rengine 1 ve 0 değeri atanmış oldu.

2. Test Cases

Test dosyası olarak, ödev zipinin içerisinde paylaşılan txt kullanıldı ve 4 tür içinde (directed list, undirected list , directed matrix, undirected matrix) ayrı ayrı olarak tüm methodlar şu şekilde test edildi,

-Hepsi için aynı dosyadan okuma yapılarak graph oluşturuldu.

-Hepsi için breadfirstsearch methodu denendi çıktıları tek tek ekrana bastırıldı.

-Hepsi için tek tek getConnectedComponentUndirectedGraph methodu denendi. Deneme try catch bloğu içerisinde yapıldı, directed metodlarda çalışmayacak olan bu method için exception yakalandı. Diğer durumlarda yani Undirected graplarda ise methodun return ettiği Graph arrayi bastırılacaktı fakat ListGraph ve MatrixGraph classlarının toString methodu olmadığı için sadece return edilebildi, bastırılmadı. Bunun yerine kaç adet bağımsız graph olduğu bastırıldı ekrana.

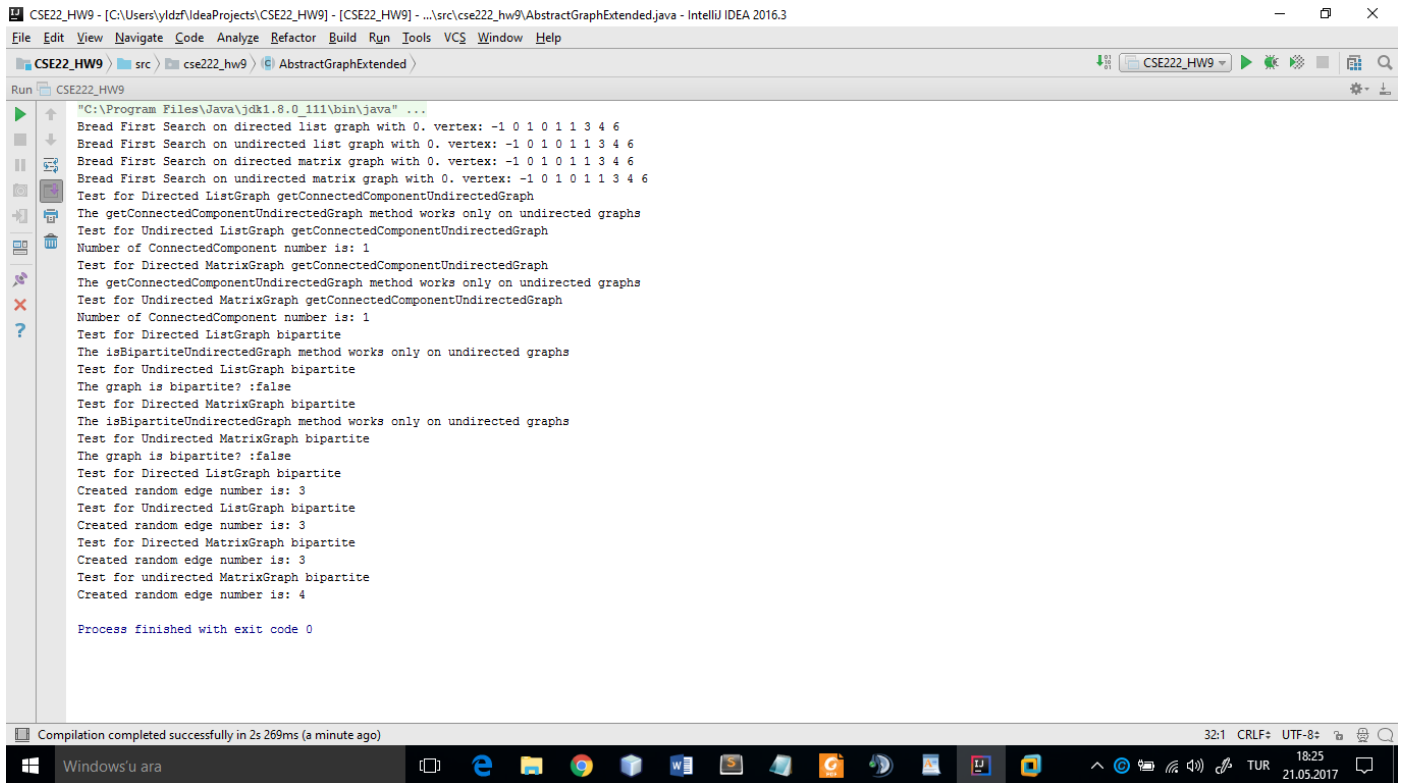
-Hepsi için graphların bipartite olup olmadığı test edildi. Deneme try catch bloğu içerisinde yapıldı, directed metodlarda çalışmayacak olan bu method için exception yakalandı. Diğer durumlarda yani Undirected graplarda ise methoddan dönen boolean değer ekrana bastırıldı.

-edgenumber 13 seçilerek, hepsi için "addRandomEdgesToGraph" methodu denendi. Random eklenen edge sayısı ekrana bastırıldı.

-Tüm graphlar kendi özelliklerinin adı ile dosyaya yazdırıldı.

Ekrana yazılırken önce testin adı sonra sonucu yazdırıldı.

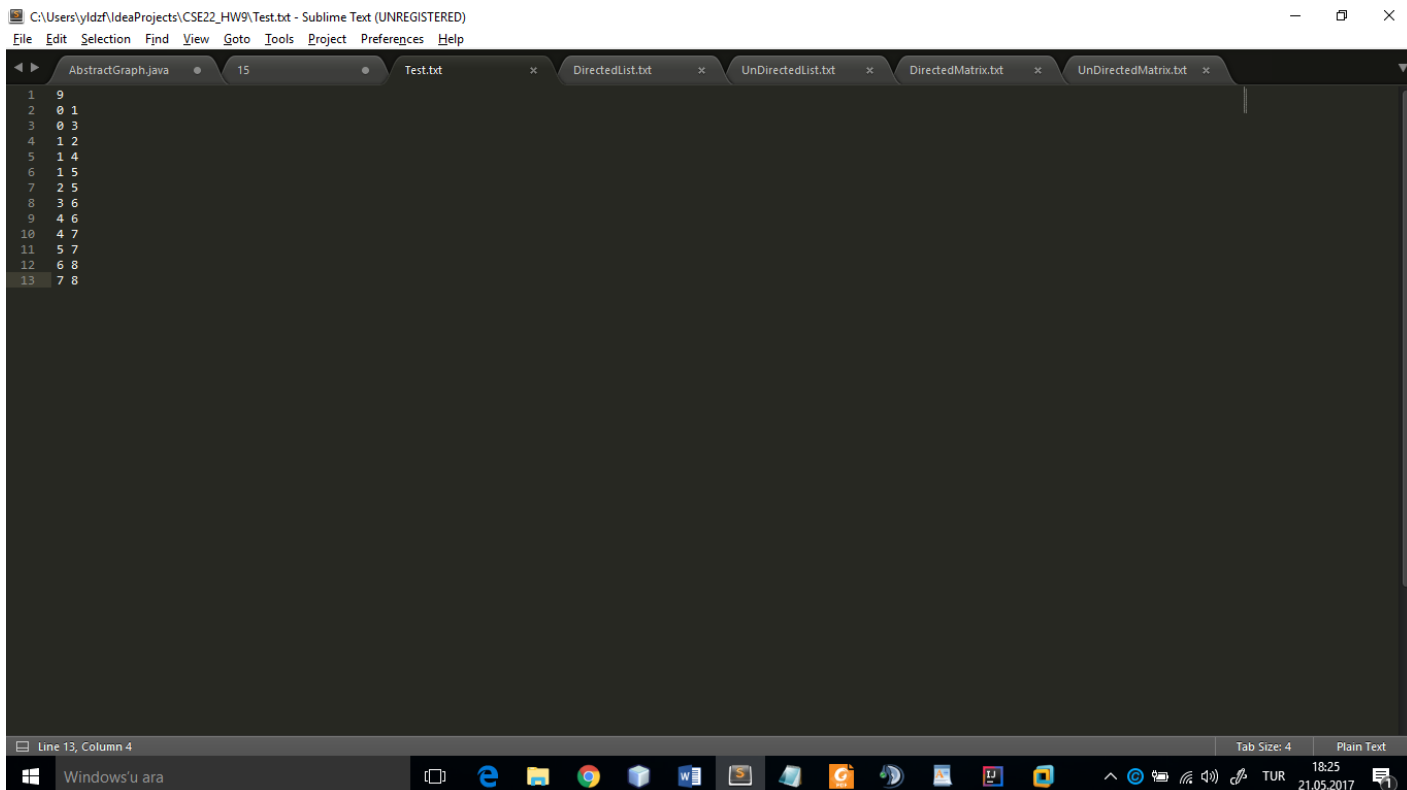
3. Running and Results



```
C:\Program Files\Java\jdk1.8.0_111\bin\java" ...
Bread First Search on directed list graph with 0. vertex: -1 0 1 0 1 1 3 4 6
Bread First Search on undirected list graph with 0. vertex: -1 0 1 0 1 1 3 4 6
Bread First Search on directed matrix graph with 0. vertex: -1 0 1 0 1 1 3 4 6
Bread First Search on undirected matrix graph with 0. vertex: -1 0 1 0 1 1 3 4 6
Test for Directed ListGraph getConnectedComponentUndirectedGraph
The getConnectedComponentUndirectedGraph method works only on undirected graphs
Test for Undirected ListGraph getConnectedComponentUndirectedGraph
Number of ConnectedComponent number is: 1
Test for Directed MatrixGraph getConnectedComponentUndirectedGraph
The getConnectedComponentUndirectedGraph method works only on undirected graphs
Test for Undirected MatrixGraph getConnectedComponentUndirectedGraph
Number of ConnectedComponent number is: 1
Test for Directed ListGraph bipartite
The isBipartiteUndirectedGraph method works only on undirected graphs
Test for Undirected ListGraph bipartite
The graph is bipartite? :false
Test for Directed MatrixGraph bipartite
The isBipartiteUndirectedGraph method works only on undirected graphs
Test for Undirected MatrixGraph bipartite
The graph is bipartite? :false
Test for Directed ListGraph bipartite
Created random edge number is: 3
Test for Undirected ListGraph bipartite
Created random edge number is: 3
Test for Directed MatrixGraph bipartite
Created random edge number is: 3
Test for undirected MatrixGraph bipartite
Created random edge number is: 4

Process finished with exit code 0
```

Compilation completed successfully in 2s 269ms (a minute ago)



```
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
```

Line 13, Column 4

C:\Users\yldzf\IdeaProjects\CSE22_HW9\DirectedList.txt - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```
1 9
2 0 1
3 0 3
4 0 7
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
```

Ekran görüntüsü kaydedildi
Ekran görüntüsü OneDrive'inıza eklendi.
OneDrive

Line 7, Column 4

Windows'u ara

C:\Users\yldzf\IdeaProjects\CSE22_HW9\UnDirectedList.txt - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

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

Ekran görüntüsü kaydedildi
Ekran görüntüsü OneDrive'inıza eklendi.
OneDrive

Line 14, Column 4

Windows'u ara

C:\Users\yldzf\IdeaProjects\CSE22_HW9\DirectedMatrix.txt - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

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

Ekran görüntüsü kaydedildi
Ekran görüntüsü OneDrive'inıza eklendi.
OneDrive

Line 1, Column 1

Windows'u ara

C:\Users\yldzf\IdeaProjects\CSE22_HW9\UnDirectedMatrix.txt - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

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

Ekran görüntüsü kaydedildi
Ekran görüntüsü OneDrive'inıza eklendi.
OneDrive

Line 14, Column 4

Windows'u ara