

ALGORİTMA ANALİZİ PROJE RAPORU

Sosyal Ağ Yapıları



Grup 1 – Mine Elif Karslıgil

MÜDAFER KAYMAK

20011093

Mudafer.kaymak@std.yildiz.edu.tr

Video Linki : <https://youtu.be/H2lh8BJupmM>

Problem Tanımı

Sosyal ağlar, bireyleri (düğümler), arkadaşlıkları (kenarlar) ile birbirine bağlayan grafikler olarak temsil edilir. Topluluk tespiti, bir sosyal ağ grafiğinde, ağın geri kalanından daha güçlü bağlantılara veya etkileşimlere sahip olan birey gruplarını tanımlamayı amaçlar. Proje kapsamında verilen bir sosyal ağ grafiğindeki toplulukları tespit etmek için bir algoritma tasarlamamız beklenmektedir. Temel hedef ağ içindeki her bir grubu (topluluğu) ve içindeki düğümleri tespit etmektir. Topluluklar içindeki düğümlerin, diğer topluluklardaki düğümlere kıyasla birbirleriyle daha güçlü bağlantılara sahip olacak şekilde bölünmenin gerçekleştirilmesidir.

Problemin Çözümü

Problemin çözümü için graph yapılarında search etme amacıyla kullanılan en popüler algoritmalarından BFS algoritması kullanılmıştır. BFS ile olası her düğümden düğüme en kısa yol hesaplanmış ve bu yollar üzerindeki kenarlarda bir sayaç tutularak hangi kenarın üzerinden ne kadar geçildiği hesaplanmıştır. Hesaplamalardan sonra eğer kullanıcıdan alınan değerler ile bir çelişki söz konusu değilse ilgili bağlantı kesilerek yeni sosyal gruplar oluşturulmuştur.

Karşılaşılan Zorluklar

Proje sırasında karşılaşılan en büyük zorluk BFS algoritmasını istençlerde bulunanlara göre uyarlamaktı. Hem t hem k girişlerine göre hareket etmek bunların yanı sıra kenarlardan en çok geçilen kenarı, graph yapısını bozmadan graphtan çıkarmak zorluklar çıkardı. Bir toplulukta bulunması gereken minimum kişiyi veren t sayısını kontrol etmek sırasında graphın belli bir işlemten sonra geçip o halini kontrol etmek ve daha sonra geriye döndürmek projenin en zorlu kısımlarındandı.

Karmaşıklık Analizi

Kodun en önemli ana algoritması BFS olduğundan, bu algoritmanın karmaşıklığını projenin karmaşıklığı olarak yazılabilir. BFS'in zaman karmaşıklığı en kötü senaryoda $O(V+E)$ 'dir. Bütün nodeları ve edgeleri gezeceği durum için bu hesap yapılmıştır. V her bir düğüm E ise her bir kenarı temsil eder.

Ekran görüntüleri

1. Örnek

```
Enter t = 2
Enter k = 2
A: E B
B: F A
C: H D
D: H C
E: F A
F: G E B
G: H F
H: G D C
Group 1: A E B F G H D C
Number of Connected Components: 1
Connected Component Sizes:
Group 1: 8 elements
src:A dest:B
Shortest Path: B <- A
src:A dest:C
Shortest Path: C <- H <- G <- F <- E <- A
src:A dest:D
Shortest Path: D <- H <- G <- F <- E <- A
src:A dest:E
Shortest Path: E <- A
src:A dest:F
Shortest Path: F <- E <- A
src:A dest:G
Shortest Path: G <- F <- E <- A
src:A dest:H
Shortest Path: H <- G <- F <- E <- A
src:B dest:A
Shortest Path: A <- B
src:B dest:C
Shortest Path: C <- H <- G <- F <- B
src:B dest:D
Shortest Path: D <- H <- G <- F <- B
src:B dest:E
Shortest Path: E <- F <- B
src:B dest:F
Shortest Path: F <- B
src:B dest:G
Shortest Path: G <- F <- B
src:B dest:H
Shortest Path: H <- G <- F <- B
src:C dest:A
Shortest Path: A <- E <- F <- G <- H <- C
src:C dest:B
Shortest Path: B <- F <- G <- H <- C
src:C dest:D
Shortest Path: D <- C
src:C dest:E
```

```
src:H dest:F
Shortest Path: F <- G <- H
src:H dest:G
Shortest Path: G <- H

Edge Matrix:
0 2 0 0 12 0 0 0
2 0 0 0 0 12 0 0
0 0 0 2 0 0 0 12
0 0 2 0 0 0 0 12
12 0 0 0 0 22 0 0
0 12 0 0 22 0 32 0
0 0 0 0 0 32 0 30
0 0 12 12 0 0 30 0
Group 1: A E B F
Group 2: C H D G
Number of Connected Components: 2
Connected Component Sizes:
Group 1: 4 elements
Group 2: 4 elements
The most visited edge F-G has been removed
Updated Adjacency List
A: E B
B: F A
C: H D
D: H C
E: F A
F: E B
G: H
H: G D C
```

Text Dosyası:

A:B,E;

B:A,F;

C:D,H;

D:C,H;

E:A,F;

F:B,E,G;

G:F,H;

H:C,D,G;

```

Shortest Path: C <- H
src:H dest:D
Shortest Path: D <- H
src:H dest:E
Couldn't reach the target.
src:H dest:F
Couldn't reach the target.
src:H dest:G
Shortest Path: G <- H

Edge Matrix:
0 2 0 0 4 0 0 0
2 0 0 0 0 4 0 0
0 0 0 2 0 0 0 4
0 0 2 0 0 0 0 4
4 0 0 0 0 6 0 0
0 4 0 0 6 0 0 0
0 0 0 0 0 0 0 6
0 0 4 4 0 0 6 0
Group 1: A E B F
Group 2: C H D G
Number of Connected Components: 2
Connected Component Sizes:
Group 1: 4 elements
Group 2: 4 elements
The most visited edge E-F has been removed
Updated Adjacency List
A: E B
B: F A
C: H D
D: H C
E: A
F: B
G: H
H: G D C

```

```

Shortest Path: D <- H
src:H dest:E
Couldn't reach the target.
src:H dest:F
Couldn't reach the target.
src:H dest:G
Shortest Path: G <- H

Edge Matrix:
0 8 0 0 6 0 0 0
8 0 0 0 0 6 0 0
0 0 0 2 0 0 0 4
0 0 2 0 0 0 0 4
6 0 0 0 0 0 0 0
0 6 0 0 0 0 0 0
0 0 0 0 0 0 0 6
0 0 4 4 0 0 6 0
Group 1: A E
Group 2: B F
Group 3: C H D G
Number of Connected Components: 3
Connected Component Sizes:
Group 1: 2 elements
Group 2: 2 elements
Group 3: 4 elements
Deletion cancelled due to value of t
A: E
B: F
C: H D
D: H C
E: A
F: B
G: H
H: G D C

```

2.Örnek

```
Enter t = 2
Enter k = 3
A: B
B: C A
C: D B
D: C E F
E: G F D
F: D H E
G: H E
H: G F
Group 1: A B C D E F G H
Number of Connected Components: 1
Connected Component Sizes:
Group 1: 8 elements
src:A dest:B
Shortest Path: B <- A
src:A dest:C
Shortest Path: C <- B <- A
src:A dest:D
Shortest Path: D <- C <- B <- A
src:A dest:E
Shortest Path: E <- D <- C <- B <- A
src:A dest:F
Shortest Path: F <- D <- C <- B <- A
src:A dest:G
Shortest Path: G <- E <- D <- C <- B <- A
src:A dest:H
Shortest Path: H <- F <- D <- C <- B <- A
src:B dest:A
```

Text Dosyası:

```
A:B;
B:A,C;
C:B,D;
D:F,E,C;
F:E,H,D;
E:D,F,G;
G:E,H;
H:F,G;
```

```
Shortest Path: B <- C <- D <- F <- H
src:H dest:C
Shortest Path: C <- D <- F <- H
src:H dest:D
Shortest Path: D <- F <- H
src:H dest:E
Shortest Path: E <- G <- H
src:H dest:F
Shortest Path: F <- H
src:H dest:G
Shortest Path: G <- H
```

Edge Matrix:

```
0 14 0 0 0 0 0 0
14 0 24 0 0 0 0 0
0 24 0 30 0 0 0 0
0 0 30 0 16 16 0 0
0 0 0 16 0 2 12 0
0 0 0 16 2 0 0 12
0 0 0 0 12 0 0 6
0 0 0 0 0 12 6 0
```

Group 1: A B C

Group 2: D E F G H

Number of Connected Components: 2

Connected Component Sizes:

Group 1: 3 elements

Group 2: 5 elements

The most visited edge C-D has been removed

Updated Adjacency List

```
A: B
B: C A
C: B
D: E F
E: G F D
F: D H E
G: H E
H: G F
```

```

src:H dest:D
Shortest Path: D <- F <- H
src:H dest:E
Shortest Path: E <- G <- H
src:H dest:F
Shortest Path: F <- H
src:H dest:G
Shortest Path: G <- H

Edge Matrix:
0 4 0 0 0 0 0 0
4 0 4 0 0 0 0 0
0 4 0 0 0 0 0 0
0 0 0 0 4 4 0 0
0 0 0 4 0 2 6 0
0 0 0 4 2 0 0 6
0 0 0 0 6 0 0 6
0 0 0 0 0 6 6 0
Group 1: A B C
Group 2: D E F H G
Number of Connected Components: 2
Connected Component Sizes:
Group 1: 3 elements
Group 2: 5 elements
The most visited edge E-G has been removed
Updated Adjacency List
A: B
B: C A
C: B
D: E F
E: F D
F: D H E
G: H
H: G F
src:A dest:B

```

```

src:H dest:D
Shortest Path: D <- F <- H
src:H dest:E
Shortest Path: E <- F <- H
src:H dest:F
Shortest Path: F <- H
src:H dest:G
Shortest Path: G <- H

Edge Matrix:
0 4 0 0 0 0 0 0
4 0 4 0 0 0 0 0
0 4 0 0 0 0 0 0
0 0 0 0 2 6 0 0
0 0 0 2 0 6 0 0
0 0 0 6 6 0 0 12
0 0 0 0 0 0 0 8
0 0 0 0 0 12 8 0
Group 1: A B C
Group 2: D E F
Group 3: G H
Number of Connected Components: 3
Connected Component Sizes:
Group 1: 3 elements
Group 2: 3 elements
Group 3: 2 elements
Deletion cancelled due to value of t
A: B
B: C A
C: B
D: E F
E: F D
F: D
G: H
H: G

```