

## Algoritma Analizi 4. Ödev

Öğrenci ad soyad:Ali Bugday

Öğrenci no:19011076

Dersin Egitmeni: Mine Elif Karslıgil

Tarih:20.12.2022

\*video link: [https://stdyildizedu-my.sharepoint.com/:f:/g/personal/ali\\_bugday\\_std\\_yildiz\\_edu\\_tr/EqRbh4O9LpFszbrPtgiZmsB97\\_iaQrx3knn2uuHoHjd4Q?e=u9nXVL](https://stdyildizedu-my.sharepoint.com/:f:/g/personal/ali_bugday_std_yildiz_edu_tr/EqRbh4O9LpFszbrPtgiZmsB97_iaQrx3knn2uuHoHjd4Q?e=u9nXVL)

### \*\*Yontem

Not: kodun içerisindeki n degerini yani grafin kaca kaclık oldugunu kod kendisi belirleyemiyor sadece onu önden ayarlamak gerekiyor.

-kod verilen dosyadaki degerleri dosya\_oku fonksiyonu ile okuyup belirlediğim bir struct dizisine kaydediyor.

-sonra bu dizideki degerleri nxn'lik bir grafa kaydediliyor.

-Algoritmanın ilk aşaması olan M'den daha az takipçisi olan kişileri tekrarlı olarak slime işini delete\_less\_followers() isimli fonksiyon ile gerçekleştiridim. bu işlemi aşağıdaki kod parçası gerçekleştiriyor.

```

while(is_there_delete(indegree, deleted, n, M))
{
    for(i=0;i<n;i++){
        if(indegree[i]<M){
            deleted[i]=1;
        }
    }
    for(i=0;i<n;i++){
        indegree[i]=0;
    }
    calculate_inorder(n, graph, M, deleted, indegree);
}

```

-yukardaki kodda is\_there\_delete m'den daha az takipçili node var mı diye control ediyor.

-calculate\_inorder silinmiş kişilere göre güncel indegreeeleri bulup tekrardan ona göre herkesin indegreelerini hesaplıyor ve indegree[] dizisine kaydediyor.

-algoritmanın ikinci kısmında dfs() isimli fonksiyonda dfs yöntemini kullanarak bütün nodeların doğrudan ve dolaylı takipçilerinin toplamını buluyor. Burda bir kişiyi doğrudan takip edenleri aslında ilk adımda hesapladığımız için toplam değeri bu değerlerden çıkartıp kişilerin dolaylı takipçilerini bulmuş oluyoruz ve bu değerleri indirect[] dizisine kaydettirdim. Aşağıdaki kod parçası bu işi yapıyor.

```

void dfs(int n, int graph[][n], int visited_by[][n], int *nodes, int *indegree, int *deleted, int *visited, int start)
{
    int j, k;
    visited[start] = 1;
    for(j=0;j<n;j++){
        if(graph[start][j]==1){
            if(visited[j]==0){//ziyaret edilmemisse
                visited[j]=1;
                if(!deleted[j]){//az takipçili değil ise
                    nodes[start]=1;
                    dfs(n, graph, visited_by, nodes, indegree, deleted, visited, j);
                    for(k=0;k<n;k++){
                        if(visited_by[j][k]==0){
                            visited_by[j][k]=nodes[k];
                        }
                    }
                    nodes[start]=0;
                }
            }
        }
    }
    /*else

```

-yukardaki fonksiyon için 1->3->5 geziyorsak nodes[n] dizisinde 1. Ve 3. Indisleri 1 yapıp, visited\_by[5][0..n]=nodes[0..n] eşitleyerek 5. kişiyi 1. ve 3. kişiler takip etti diyoruz.

## **\*\*Uygulama**

\*M=3, X=3, Y=4 için:

```
D:\yak2kl2 ali\Documents\_c3nc3_sinif_g3z\Algorithm_analyze\hw\odev_4\hw4\190
M degerini giriniz:3
X degerini giriniz:3
Y degerini giriniz:4
1. Normal mod icin: 1
2. Detayl2 mod icin: 2
2

Baslangic durumunda noktalarin indegree degerleri:
2-2-2-1-1-2-1-4-1-2-2-2-
Verilen M degeri i0in silinmeyen kisiler ve indegreeleri:

Verilen X ve Y degerleri i0in infulencer olanlar2n bilgileri:

-----
Process exited after 6.904 seconds with return value 0
Press any key to continue . . .
```

-Kimseyi influencer olarak tanımlamadı. Başlangıç durumunda indegree değeri M=3'den büyük olan tek bir nokta var o da diğer noktaların silinmesiyle sıfırlanır.

\*M=2, X=3, Y=3:

D:\yak<sup>2</sup>kl<sup>2</sup> ali\Documents\\_c<sup>3</sup>nc<sup>3</sup>\_sinif\_g<sup>3</sup>z\Algorithm\_analyze\hw\odev\_4\hw4

Y degerini giriniz:3

1. Normal mod icin: 1

2. Detayl<sup>2</sup> mod icin: 2

2

Baslangic durumunda noktalarin indegree degerleri:

2-2-2-1-1-2-1-4-1-2-2-2-

Verilen M degeri i<sup>2</sup>in silinmeyen kisiler ve indegreeleri:

Michael Jordan

: 2

Stephen Boyd

: 2

Kalyanmoy Deb

: 2

David Johnson

: 1

Scott Kirkpatrick

: 1

Lieven Vandenberghe

: 2

Fabian Pedregosa

: 1

Jorge Nocedal

: 4

Clifford Stein

: 1

Stephen Wright

: 2

Philippe Salembier

: 2

Robert Stevenson

: 2

Verilen X ve Y degerleri i<sup>2</sup>in infulencer olanlar<sup>2</sup>n bilgileri:

Lieven Vandenberghe

direct:2 + indirect:9 total11

Jorge Nocedal

direct:4 + indirect:7 total11

Stephen Wright

direct:2 + indirect:9 total11

Philippe Salembier

direct:2 + indirect:9 total11

Robert Stevenson

direct:2 + indirect:9 total11

[illegible]

```
M degerini giriniz:1
X degerini giriniz:2
Y degerini giriniz:4
1. Normal mod icin: 1
2. Detayl2 mod icin: 2
2
```

2-2-2-1-1-2-1-4-1-2-2-2-

Verilen M degeri iin silinmeyen kisiler ve indegreeeleri:

Michael Jordan

: 2

Stephen Boyd

$$: 2$$

Kalvanmoy Deb

$$: 2$$

David Johnson

$$: 1$$

Scott Kirkpatrick

$$: 1$$

Lieven Vandenberghe

: 2

Fabian Pedregosa

$$: 1$$

Jorge Nocedal

: 4

Clifford Stein

$$\vdots 1$$

Stephen Wright

: 2

Philippe Salembier

2

Robert Stevenson

2

\_\_\_\_\_

Lieven Vandenberghe

```
direct:2 + indirect:9 total11
```

Jorge Nocedal

```
direct:4 + indirect:7 total11
```

Stephen Wright

```
direct:2 + indirect:9 total11
```

Philippe Salembier

```
direct:2 + indirect:9 total11
```

Robert Stevenson

```
direct:2 + indirect:9 total11
```

\_\_\_\_\_

**\*\*sonuc**

**\*dogrudan takipcileri bulmanın karmaşıklığı:**

```
while(is_there_delete(indegree, deleted, n, M)) → K
{
    for(i=0; i<n; i++){
        if(indegree[i]<M){
            deleted[i]=1;
        }
    }
    for(i=0; i<n; i++){
        indegree[i]=0;
    }

    calculate_inorder(n, graph, M, deleted, indegree);
}
```

$O(Kn^2)$

$\rightarrow n^2$

-yukarda k while döngüsünün m'den az takipcisi olmayan nokta kalıncaya kadar dönmesi.

-calculate order kendi içerisinde  $n^2$  defa çalışıyor.

**\*dfs:**

-For döngüsü içerisinde bütün noktalar için dfs çağrılır:

```
for(i=0; i<n; i++){
    for(j=0; j<n; j++){
        visited[j]=0;
        nodes[j]=0;
    }

    dfs(n, graph, visited_by, nodes, indegree, deleted, visited, i);
}
```

$\rightarrow n$

-dfs de kendi icerisinde  $O(N+E)$  defa calisir.

-toplamda karmaşıklığımız  $O(N*(N+E))$  olur.