



**Algoritma Analizi 3. Ödevi**

**Muhammet Kayra Bulut**

**20011901**

## Yöntem

Dosyadan okuma işlemini gerçekleştirdikten sonra, BFS algoritması kullandım. Bu algoritmayı kullanmak için, “enqueue” ve “dequeue” işlemleri yapabileceğim bir “queue” yapısı oluşturdum ve ilgili fonksiyonları tanımladım. Verilen kişilerin değerini tutabilmek için komşuluk listesi oluşturdum ve bu listede “KISI” isimli struct veri yapısını tuttum. KISI isimli structımın içinde de “ID” isimli structum vardı ki bu da takip edilen kullanıcıların listesini tutan bir veri yapısı. Sonrasında ilk olarak “indegreeHesapla” isimli in-degree hesaplama fonksiyonunu yazdım. Bu fonksiyon basit olarak, her elemanın komşularının değerlerini birer birer artırıyor. Sonrasında “indegreeEksilt” isimli fonksiyonu tanımladım ve bu fonksiyonun yaptığı şey, verilen eşik değerden daha düşük değere sahip elemanları silip, aynı zamanda in-degree değerlerini azaltmak ve azaltırken de yine eşik değer altına düşen elemanları rekürsif olarak silmek. BFS fonksiyonum da ilgili düğümden yol olan tüm düğümleri buluyor. Bu şekilde de komşu sayısını hesaplayabiliyorum.

## Uygulama

```
M degerini giriniz
1
X degerini giriniz
2
Y degerini giriniz
3
Normal Mod Icin 1
Detay Mod Icin 2
Cikis icin 0
1
ID->6--ISIM->Lieven--SOYISIM->Vandenberghe
ID->8--ISIM->Jorge--SOYISIM->Nocedal
ID->10--ISIM->Stephen--SOYISIM->Wright
ID->11--ISIM->Philippe--SOYISIM->Salembier
ID->12--ISIM->Robert--SOYISIM->Stevenson
```

Normal Mod

```
Normal Mod Icin 1
Detay Mod Icin 2
Cikis icin 0
2
Baslangic in-degree degerleri
1. id -> 2
2. id -> 2
3. id -> 2
4. id -> 1
5. id -> 1
6. id -> 2
7. id -> 1
8. id -> 4
9. id -> 1
10. id -> 2
11. id -> 2
12. id -> 2
Elenmeyen dugumlerin in-degree degerleri
1. id -> 2
2. id -> 2
3. id -> 2
4. id -> 1
5. id -> 1
6. id -> 2
7. id -> 1
8. id -> 4
9. id -> 1
10. id -> 2
11. id -> 2
12. id -> 2
Influencer Bilgileri
ID->6--ISIM->Lieven--SOYISIM->Vandenberghe
Indegree -> 2 --- Komsu Sayisi--->11
ID->8--ISIM->Jorge--SOYISIM->Nocedal
Indegree -> 4 --- Komsu Sayisi--->11
ID->10--ISIM->Stephen--SOYISIM->Wright
Indegree -> 2 --- Komsu Sayisi--->11
ID->11--ISIM->Philippe--SOYISIM->Salember
Indegree -> 2 --- Komsu Sayisi--->11
ID->12--ISIM->Robert--SOYISIM->Stevenson
Indegree -> 2 --- Komsu Sayisi--->11
```

### Detay Mod

```
Normal Mod Icin 1
Detay Mod Icin 2
Cikis icin 0
0
Cikis yapildi
```

### Çıkış

## Sonuç

BFS() -> Maksimum V kadar enqueue yapabilir, bundan dolayı dış döngü V defa dönecek. Maksimum E sayısı kadar komşuluğa bakabilir. Bundan dolayı da iç döngü E defa dönecek. İç içe olduğu için karmaşıklık  **$O(V \cdot E)$**

initilizeMatrix() -> Dış döngü de iç döngü de V kadar döner. Bundan dolayı  **$O(V^2)$**

initilizeArray() -> Tek döngü V defa döner.  **$O(V)$**

indegreeEksilt()-> Döngü E kadar dönebilir ve kendini de en fazla V kadar çağırabilir. Bu yüzden  **$O(V \cdot E)$**

indegreeHesapla() -> En fazla E kadar döner. Bundan Dolayı  **$O(E)$**

kisiYazdir()-> Belli sayıda işlem vardır. Bundan dolayı  **$O(1)$**

kisiOlustur()->Belli sayıda işlem vardır. Bundan dolayı  **$O(1)$**

enqueue()->Belli sayıda işlem vardır. Bundan dolayı  **$O(1)$**

dequeue()->Belli sayıda işlem vardır. Bundan dolayı  **$O(1)$**

readFile()-> Dış döngü V defa iç döngüyse E defa döner.  **$O(V \cdot E)$**