

Algoritma Analizi Odev1

Soru-2

```
package odev2;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.util.Random;

public class Degerlendirme {

    //tabloyu rasgele bir biçimde oluştur
    public void TabloOlustur(int k,int users){
        File f=new File("C://Users/heydar/workspace/Alg-Anal-Odev1/src/odev2/tablo.txt");
        PrintWriter writer = null;
        try {
            writer=new PrintWriter(f,"UTF-8");
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (UnsupportedEncodingException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        //header satırını ekle
        String line="";
        for(int i=0;i<k;i++){
            line=line+" "+"header-"+(i+1);
            writer.print(line);
            line="";
        }
        //1 ve 0 satırlarını ekle
        for(int i=0;i<users;i++){
            String satir="";
            writer.println();
            for(int j=0;j<k;j++){
                Random r=new Random();
                satir=satir+(r.nextInt(2));
                writer.print(satir);
                satir="";
            }
        }
        writer.close();
    }

    public String[] Kumele(int t,int k){
        BufferedReader br = null;
        String line = null;
        String[] headers=new String[2];//bulunacak olan headerları tut
```

```

//ikili1 ve ikili2 değişkeni seçilen en fazla headerların indexlerini tutuyor
//eb_ikili seçilmiş ikilinin seçim sayısını tutuyor
int ikili1=0,ikili2=0,eb_ikili=0,kisi_sayisi = 0,gecici_kisi_sayisi=0;
//butun ikilileri test et
for(int i=0;i<t-1;i++){
    for(int j=i+1;j<t;j++){
        try {
            br=new BufferedReader(new FileReader (" C://Users/
heydar/workspace/Alg-Anal-Odev1/src/odev2/tablo.txt"));
            line=br.readLine();
        }
        catch (FileNotFoundException e) {}
        catch (IOException e) {}
        //test edilen ikilinin seçim sayısını tutuyor
        int ikili=0;
        //bir ikili seçildikten sonra butun dosyayı gez
        while(line!=null){
            try {
                line=br.readLine();
                if(line==null)break;
                //test edilen ikililerin o kişiye göre degerlerini al
                char c1=line.charAt(i);
                char c2= line.charAt(j);
                if(c1=='1' && c2=='1'){
                    ikili++;
                    gecici_kisi_sayisi++;
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        //eger test edilen ikili en fazla seçilmişse
        //en fazla seçilen olarak bu ikiliyi ve bu ikililerin indexlerini al
        if(eb_ikili<ikili){
            ikili1=i;
            ikili2=j;
            eb_ikili=ikili;
            kisi_sayisi=gecici_kisi_sayisi;
        }
        gecici_kisi_sayisi=0;
    }
}
System.out.println(kisi_sayisi+" Kisi bu iki urunu almıştır.");
headers[0]="header-"+(ikili1+1);
headers[1]="header-"+(ikili2+1);
return headers;
}

```

```

public static void main(String[] args) {
    //5 urunu,ve 100 musteriye sahip bir tablo oluşturun
    new Degerlendirme().TabloOlustur(5,100);
}

```

```

        //en fazla alınmış iki ürünü bul
        String[] s=new Degerlendirme().Kumele(5,100);
        for (String string : s) {
            if(string!=null)
                System.out.println(string);
        }
    }
}

```

Örnek çıktısı:

Tablo.txt

```

header-1 header-2 header-3 header-4 header-5
10100
01101
11001
11001
11100
01001
00011
11111
11100
11011

```

Konsole Çıktısı.

```

6 Kisi bu iki ürünü almıştır.
header-1
header-2

```

Bu algoritmanın çözüm yontemi her satır için header'ları ikili gruplar haline getirerek hesaplamaktır.Header sayısı artıkça headerların ikili grup sayısında yaklaşık öncekinin iki katı kadar artmaktadır.Buradan algoritmanın karmaşıklığının $O(2^n)$ olduğunu söyleyebiliriz.(n header sayısı)