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){
		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<k;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();
	}
	//uygun eşik degerine karşılık gelen header'ları geri dondur
	public String[] getHeader(int t,int k){
		BufferedReader br = null;
		String line;
		String[] headers=new String[k];
		try {
			br=new BufferedReader(new FileReader ("C://Users/heydar/workspace/
				Alg-Anal-Odev1/src/odev2/tablo.txt"));
			line=br.readLine();
		} catch (FileNotFoundException e) {
			e.printStackTrace();
		} catch (IOException e) {
```

```
                e.printStackTrace();
            }
            for(int i=0;i<k;i++){
                int esik=0;
                for(int j=0;j<k;j++){
                    try {
                        line=br.readLine();
                        char c=line.charAt(i);
                        if(c=='1'){
                            esik++;
                        }
                    } catch (IOException e){
                        e.printStackTrace();
                    }
                }
                try {
                    br=new BufferedReader(new FileReade r("C://Users/heydar/
                    workspace/Alg-Anal-Odev1/src/odev2/tablo.txt"));
                    line=br.readLine();
                } catch (FileNotFoundException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                if(esik>t){
                    headers[i]="header-"+(i+1);
                }

            }
            return headers;
        }
        public static void main(String[] args) {
                //6 headerı olan bir tablo oluştur
                new Degerlendirme().TabloOlustur(6);
                //eşik degeri 2'den yuksek olan headerları geri dondur
                String[] s=new Degerlendirme().getHeader(2, 6);
                for (String string : s) {
                    if(string!=null)
                    System.out.println(string);
                }
        }
}
```

Bu programda kendisine gonderilen k parametresine göre k sayisinda header'a ve k sayisinda bir ve sıfırlardan oluşan satıra sahip bir tablo.txt dosyası oluşturumakta ve parametre olarak gonderilen eşik degerine karşılık gelen headerlar bulunarak geri dondurulur.Bu işlemi gerçekleştiren getHeader(int t,int k) metodunun kodunun karmaşıklıgı k degerine gore değişmektedir.Öyleki;

```
        for(int i=0;i<k;i++){
                int esik=0;
                for(int j=0;j<k;j++){
                    try {
                        line=br.readLine();
                        char c=line.charAt(i);
                        if(c=='1'){
                            esik++;
                        }
                    } catch (IOException e){
```

```
                        e.printStackTrace();
                    }
                }
                try {
                    br=new BufferedReader(new FileReade r("C://Users/heydar/__
                  workspace/Alg-Anal-Odev1/src/odev2/tablo.txt"));
                    line=br.readLine();
                } catch (FileNotFoundException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
                if(esik>t){
                    headers[i]="header-"+(i+1);
                }

            }
```

kodunda k'nın değerine baglı olarak iki nested for dongüsüne sahiptir. Tablodaki herbir bir ve sıfır degerlerinin kontrolü yapılmaktadır.k'nın çok buyuk degerlerinde diğer işlemleri ihmal edersek bu kodun karmaşıklığı O(k2)'dir.(k-tablodaki header sayısı).

Örnek çıktısı:

Tablo.txt

```
header-1  header-2  header-3  header-4  header-5  header-6
110100
100101
011010
000111
000011
001000
```

Konsole Çıktısı.(2 eşik değerinden buyuk olan headerler)

```
header-4
header-5
header-6
```