Bu programda main metodu hariç 4 metot kullanıldı. Bunlar; random, duzFragmanlar, tumleyenFragmanlar ve hizalama metodları.

random metodu;

```
public static String[] random(int y) {
		String alfabe = "ACGT";
		Random random = new Random();
		String[] sekans = new String[y];
		for(int i=0; i<y;i++) {
		String a = Character.toString(alfabe.charAt(random.nextInt(alfabe.length())));
		sekans[i] = a;
		}
		return sekans ;
	}
```

Bu metot sekansın rastgele bir şekilde üretilmesi istendiğinde main metodu tarafından çağırılıyor. Döngü içerisinde ACGT karakterlerinden birini rastgele olarak seçiyor ve bunu sekans adlı dizinin sıradaki elemanına atıyor. Bu sekans bizim ana sekansımız.

duzFragmanlar metodu;

```
public static String[][] duzFragmanlar(int n, int l, String[] sekans) {

		int dongu = sekans.length-l+1;
		Random random = new Random();
		String[][] fragmanlar = new String[n][l];
		for(int i=0;i<n;i++) {
			int a = random.nextInt(dongu);
			for(int j=0;j<l;j++,a++) {
			fragmanlar[i][j] = sekans[a];
			}
		}
		return fragmanlar;
	}
```

Bu metot parametre olarak fragmanların sayısını (n), fragmanların uzunluğunu (l) ve ana sekansı alıyor. Rastgele seçilen bir sayıyı o sayıdan (indisten) başlayarak sekanstan l uzunlukta fragman oluşturuyor. Ve bunu n defa yapıyor. Ve bunu iki boyutlu fragmanlar dizisine (matrise) kaydediyor.

tumleyenFragmanlar metodu;

```
public static String[][] tumleyenFragmanlar(String[][] fragmanlar){
		final int N = fragmanlar.length;
		final int L = fragmanlar[0].length;
		String[][] ters_fragmanlar = new String[N][L];
		for(int i=0;i<N;i++) {
			for(int j=0;j<L;j++) {
				switch(fragmanlar[i][j]) {
				case "A":
					ters_fragmanlar[i][j] = "T";
					break;
				case "C":
```

```
                                        ters_fragmanlar[i][j] = "G";
                                        break;
                                case "G":
                                        ters_fragmanlar[i][j] = "C";
                                        break;
                                case "T":
                                        ters_fragmanlar[i][j] = "A";
                                        break;
                                default:
                                        ters_fragmanlar[i][j] = "A";
                                }
                        }
                }
                String[][] tumleyen_fragmanlar = new String[N][L];
                for(int i=0;i<N;i++) {
                        for(int j=0;j<L;j++) {
                                tumleyen_fragmanlar[i][j] = ters_fragmanlar[i][L-j-1];
                        }
                }
                return tumleyen_fragmanlar;
        }
```

Bu metot da parametre olarak iki boyutlu String dizisi alıyor. Yani az önce oluşturduğumuz fragmanlar matrisini. İlk önce bu matristeki nükleotidleri eşleriyle değiştiriyor. Sonra bu diziyi tersten yazarak fragmanların tümleyenlerini üretiyor. return değeri olarak da bu matrisi veriyor.

hizalama metodu;

```
public static int[][] hizalama(String[][] fragmanlar, String[][] tumleyen_fragmanlar){
                final int N = fragmanlar.length;
                final int L = fragmanlar[0].length;
                int[][] skor_matris = new int[N][N];
                int[][] yerel_hizalama = new int[L+1][L+1];
                for(int i=0;i<=L;i++) {
                yerel_hizalama[0][i] = 0;
                yerel_hizalama[i][0] = 0;
                }
                for(int k=0;k<N;k++) {
                for(int m=1;m<N;m++) {
                for(int i=0;i<L;i++) {
                        for(int j=0;j<L;j++) {
        if(fragmanlar[k][i] == fragmanlar[m][j]) {
        yerel_hizalama[i+1][j+1] = yerel_hizalama[i][j] + hamdi.eslesme;
        }
        else {
yerel_hizalama[i+1][j+1] = Math.max((Math.max((yerel_hizalama[i][j+1]-hamdi.indel),
(yerel_hizalama[i+1][j]-hamdi.indel))),yerel_hizalama[i][j]-hamdi.eslesmeme);
                                                }}}
                                skor_matris[k][m] = yerel_hizalama[L][L];
                        }}
                for(int k=0;k<N;k++) {
                for(int m=1;m<N;m++) {
                for(int i=0;i<L;i++) {
                for(int j=0;j<L;j++) {
        if(tumleyen_fragmanlar[k][i] == tumleyen_fragmanlar[m][j]) {
```

```
        yerel_hizalama[i+1][j+1] = yerel_hizalama[i][j] + hamdi.eslesme;
        }
        else {
yerel_hizalama[i+1][j+1] = Math.max((Math.max((yerel_hizalama[i][j+1]-hamdi.indel),
(yerel_hizalama[i+1][j]-hamdi.indel))),yerel_hizalama[i][j]-hamdi.eslesmeme);
                                                        } } }
                                        skor_matris[m][k] = yerel_hizalama[L][L];
                                } }
        return skor_matris;
                }
```

Bu metot parametre olarak hem fragmanları hem de tümleyen fragmanları alıyor. İlk önce yerel hizalama matrisi oluşturuyor ve ilk satıra ve ilk sütuna 0 atıyor. fragmanlar[0] ve fragmanlar[1] adlı fragmandan hizalamaya başlıyor. İlgili nükleotidler eşitse sol ve yukarıdaki matris elemanına eslesme skorunu ekliyor ve yerel_hizalama matrisinin ilgili indisine kaydediyor. Eşit değiller ise 3 olasılıktan en yüksek skor olarak geri döneni ilgili indise yazıyor. Bu olasılıklar yukarıdan indel, soldan indel ve sol yukarıdan mismatch. Bu işlemleri hem fragmanlar hem de tümleyen fragmanlar için yapıyor. return değeri olarak da skor_matrisini veriyor.

main metodu;

```
public static void main(String[] args) throws FileNotFoundException {
                Scanner girdi = new Scanner(System.in);
                System.out.println("Kac adet fragman istiyorsunuz?");
                final int N = girdi.nextInt();
                System.out.println("Fragmanlar kac nükleotit uzunlugunda olacak?");
                final int L = girdi.nextInt();
                System.out.println("Sekansı kendiniz girmek istiyorsanız 1,"
                                + "\nRastgele üretilmesini istiyorsanız herhangi bir karakter giriniz.");
                int secenek = girdi.nextInt();
                System.out.println("Eslesme odul puanini giriniz.");
                hamdi.eslesme = girdi.nextInt();
                System.out.println("Eslesmeme ceza puanini giriniz.");
                hamdi.eslesmeme = Math.abs(girdi.nextInt());
                System.out.println("Indel ceza puanini giriniz.");
                hamdi.indel = Math.abs(girdi.nextInt());
                System.out.println("Skor matrisinde kabul edilen en kucuk skoru giriniz.");
                hamdi.T = girdi.nextInt();

                File dosya = new File("dosya.txt");
                PrintWriter output = new PrintWriter(dosya);

                final String[][] fragmanlar;

                if(secenek == 1) {
                System.out.print("Sekansinizi giriniz");
                String girdi_sekans = girdi.next();
                int a = girdi_sekans.length();
                girdi_sekans = girdi_sekans.toUpperCase();
                System.out.println("Sekansiniz " + a + " nukleotid uzunlukludur.");
                System.out.println(girdi_sekans);
                String[] sekans = new String[a];
                for(int i=0;i<a;i++) {
                sekans[i] = Character.toString(girdi_sekans.charAt(i));
                }

                fragmanlar = duzFragmanlar(N,L,sekans);
```

```
}
else{
		System.out.println("Ana sekansiniz kac nukleotid uzunlugunda olacak?");
		int random_sekans_uzunlugu = girdi.nextInt();
		String[] random_sekans = random(random_sekans_uzunlugu);
		fragmanlar = duzFragmanlar(N,L,random_sekans);
		System.out.println("Sekansiniz:");
		System.out.print("[");
		for (int i=0;i<random_sekans.length;i++) {
			System.out.print(random_sekans[i]);
		}
		 System.out.print("]");
}
System.out.println();

String [][] tumleyen_fragmanlar = tumleyenFragmanlar(fragmanlar);

for(int i=1;i<((fragmanlar[0].length)/2);i++) {
		System.out.print(" ");
		}
		System.out.print("Read");

		for(int i=0;i<fragmanlar[0].length;i++) {
		System.out.print(" ");
		}
		System.out.println("Read*");
		for (int i = 0; i<N; i++) {
			System.out.print("[");
		   for (int j = 0; j<L; j++) {
		      System.out.print(fragmanlar[i][j]);
		   }
		   System.out.print("] ");
		   System.out.print(" [");
		   for (int j = 0; j<L; j++) {
		      System.out.print(tumleyen_fragmanlar[i][j]);
		   }
		   System.out.print("]");
		   System.out.println();
		}
		System.out.println();

		int[][] skor_matris = hizalama(fragmanlar,tumleyen_fragmanlar);
		String[][] str_matris = new String[N][N];
		for(int i=0;i<N;i++) {
			for(int j=0;j<N;j++) {
			str_matris[i][j] = String.valueOf(skor_matris[i][j]);
			}
		}
		for(int i=0;i<N;i++) {
			for(int j=0;j<N;j++) {
				if(skor_matris[i][j] < hamdi.T) {
					str_matris[i][j] = " ";
				}
				if(i==j) {
					str_matris[i][j]= "*";
				}
			}
		}
		output.printf("    ");
		for(int j=0;j<N;j++) {
```

```
                    output.printf("%4s", j);
                }
                output.printf("\n");
                for(int i=0;i<N;i++) {
                    output.printf("%4s", i);
                    for(int j=0;j<N;j++) {
                        output.printf("%4s", str_matris[i][j]);
                    }
                    output.println();
                }
                output.close();

            girdi.close();
        }
```

main metodunun ilk kısmında klavyeden gerekli girdileri alıyor. Sekansı, Read ve Read* fragmanlarını konsola yazdırıyor. Ardından tüm metotlar sırayla kullanıyor. En son hizalama metotunun return değerini String bir matrise dönüştürüyor. Sor matrisindeki değerler kabul edilen en küçük değerden daha küçükse boş bırakılıyor. Üst matris fragmanların skorlarını, alt matris de tümleyen fragmanların skorlarını karşılıyor. Ve bunları önceden yaratmış olduğu dosya.txt adlı dosyaya yazdırıyor.

Kodun genel olarak planı da şu şekildedir;

```
import java.io.*;
import java.util.*;

public class hamdi {

    public static int T;
    public static int eslesme;
    public static int eslesmeme;
    public static int indel;

public static void main(String[] args) throws FileNotFoundException {

public static String[] random(int y) {

public static String[][] duzFragmanlar(int n, int l, String[] sekans) {

public static String[][] tumleyenFragmanlar(String[][] fragmanlar){

public static int[][] hizalama(String[][] fragmanlar, String[][] tumleyen_fragmanlar){

}
```

```
Kac adet fragman istiyorsunuz?
20
Fragmanlar kac nükleotit uzunlugunda olacak?
8
Sekansı kendiniz girmek istiyorsanız 1,
Rastgele üretilmesini istiyorsanız herhangi bir karakter giriniz.
2
Eslesme odul puanini giriniz.
2
Eslesmeme ceza puanini giriniz.
2
Indel ceza puanini giriniz.
1
Skor matrisinde kabul edilen en kucuk skoru giriniz.
8
Ana sekansiniz kac nukleotid uzunlugunda olacak?
70
Sekansiniz:
[ATTTGACCGTTATTACGGCAGGTTGGGAAAATTCGATCGGCCATCCTTTTAACCGGACCGCGAAGCTCCC]
   Read        Read*
[CGATCGGC]  [GCCGATCG]
[CTTTTAAC]  [GTTAAAAG]
[TTTGACCG]  [CGGTCAAA]
[GCAGGTTG]  [CAACCTGC]
[ACCGCGAA]  [TTCGCGGT]
[TCGGCCAT]  [ATGGCCGA]
[CGATCGGC]  [GCCGATCG]
[CATCCTTT]  [AAAGGATG]
[ATTACGGC]  [GCCGTAAT]
[ATTCGATC]  [GATCGAAT]
[TTTTAACC]  [GGTTAAAA]
[TCGATCGG]  [CCGATCGA]
[CGATCGGC]  [GCCGATCG]
[GCCATCCT]  [AGGATGGC]
[TTGACCGT]  [ACGGTCAA]
[TTCGATCG]  [CGATCGAA]
[TTGACCGT]  [ACGGTCAA]
[TCCTTTTA]  [TAAAAGGA]
[CCGGACCG]  [CGGTCCGG]
[TTGGGAAA]  [TTTCCCAA]
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	*						16					13	16			10				
1		*									13							10		
2			*								9				13	8	13			
3				*																
4					*															
5						*														
6	16						*		8			13	16			10				
7								*						10				10		
8	8						8		*	8			8							
9									8	*		10				13				
10		13	9								*									
11	13						13			10		*	13			13				
12	16						16		8			13	*			10				
13								10						*						
14			13												*	9	16			
15	10		8				10			13		13	10		9	*	9			
16			13												16	9	*			
17		10						10										*		
18																			*	
19																				*

Programa ekranda verilen değerler girildiğinde konsolda çıkan sekanslar ve fragmanlar. Ve dosya.txt dosyasındaki skor matrisi.