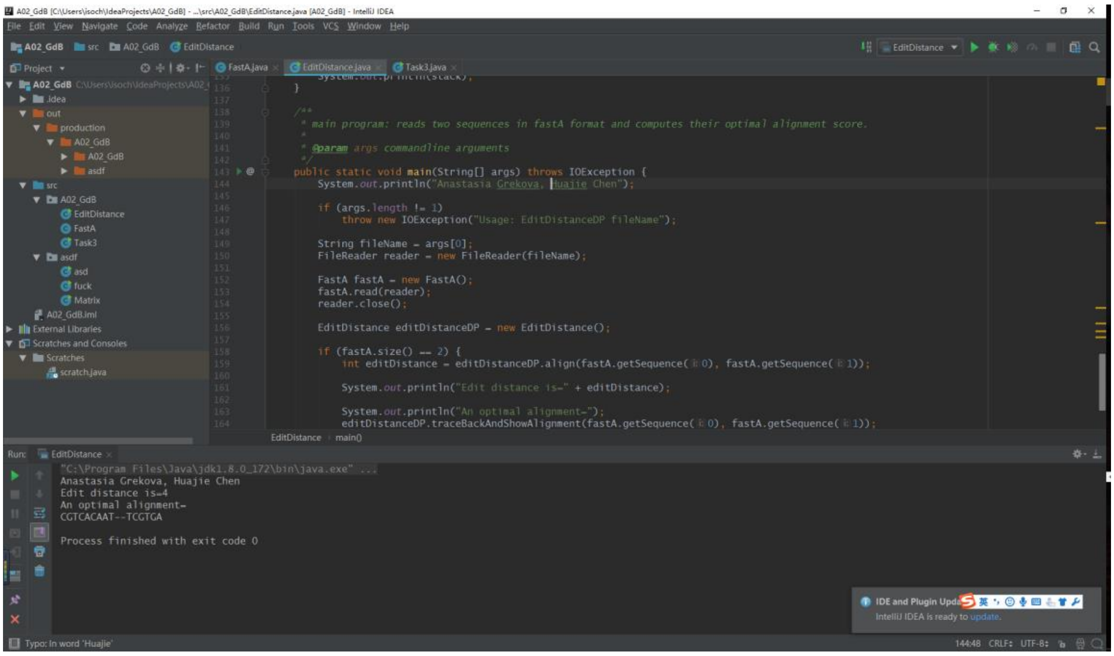
```java
package A02_GdB;

import java.io.FileReader;
import java.io.IOException;

/**
 * Compute edit distance using dynamic programming
 * Anastasia Grekova, Huajie Chen
 * 27.04.2018
 */
public class EditDistance {

    /**
     * computes the edit distance between two sequences using dynamic programming
     *   This method sets up and fills the dynamic programming matrix
     *
     * @param word1 first sequence
     * @param word2 second sequence
     * @return edit distance
     */
    public int align(String word1, String word2) {
        // PLEASE IMPLEMENT (first assignment task)
        int len1 = word1.length();
        int len2 = word2.length();

        // len1+1, len2+1, because finally return dp[len1][len2]
        int[][] dp = new int[len1 + 1][len2 + 1];

        for (int i = 0; i <= len1; i++) {
            dp[i][0] = i;
        }

        for (int j = 0; j <= len2; j++) {
            dp[0][j] = j;
        }

        //iterate though, and check last char
        for (int i = 0; i < len1; i++) {
            char c1 = word1.charAt(i);
            for (int j = 0; j < len2; j++) {
                char c2 = word2.charAt(j);

                //if last two chars equal
                if (c1 == c2) {
                    //update dp value for +1 length
                    dp[i + 1][j + 1] = dp[i][j];
                } else {
                    int replace = dp[i][j] + 1;
                    int insert = dp[i][j + 1] + 1;
                    int delete = dp[i + 1][j] + 1;

                    int min = replace > insert ? insert : replace;
                    min = delete > min ? min : delete;
                    dp[i + 1][j + 1] = min;
                }
            }
        }

        return dp[len1][len2];
    }


    /**
     * perform traceback and print an optimal alignment to  the console (standard output)
     *  This method assumes that the method align has already been run and that the
dynamic programming
     *  matrix has been computed and is stored in the class
     */

    public void traceBackAndShowAlignment(String word1, String word2) {
        int len1 = word1.length();
        int len2 = word2.length();

        // 码好最初的行列
        int[][] dp = new int[len1 + 1][len2 + 1];

        for (int i = 0; i <= len1; i++) {
            dp[i][0] = i;
        }

        for (int j = 0; j <= len2; j++) {
            dp[0][j] = j;
        }

        //计算矩阵
        for (int i = 0; i < len1; i++) {
            char c1 = word1.charAt(i);
            for (int j = 0; j < len2; j++) {
```

```
        char c2 = word2.charAt(j);

        //if last two chars equal
        if (c1 == c2) {
            //update dp value for +1 length
            dp[i + 1][j + 1] = dp[i][j];
        } else {
            int replace = dp[i][j] + 1;
            int insert = dp[i][j + 1] + 1;
            int delete = dp[i + 1][j] + 1;

            int min = replace > insert ? insert : replace;
            min = delete > min ? min : delete;
            dp[i + 1][j + 1] = min;
        }
    }
}

String stack = "";
for(int i = 1, j= 1; i <= len1 || j <= len2;){
    char c1 = word1.charAt(len1 - i);
    char c2 = word2.charAt(len2 - j);

    if(c1 == c2){
        stack += c1;
        ++i;
        ++j;
    }else {
        int replace = dp[len1 - i][len2 - j];
        int insert = dp[len1 - i + 1][len2 - j];
        int delete = dp[len1 - i][len2 - j + 1];
        if(dp[len1 - i + 1][len2 - j + 1] - 1 == replace){
            stack += c2;
            ++i;
            ++j;
        }else if(dp[len1 - i + 1][len2 - j + 1] - 1 == insert){
            stack += "-";
            ++j;
        }else if(dp[len1 - i + 1][len2 - j + 1] - 1 == delete){
            stack += "";
            ++i;
        }
    }
}
```

```
        stack = new StringBuilder(stack).reverse().toString();

        System.out.println(stack);
    }

    /**
     * main program: reads two sequences in fastA format and computes their optimal alignment score.
     *
     * @param args commandline arguments
     */
    public static void main(String[] args) throws IOException {
        System.out.println("Huajie Chen");

        if (args.length != 1)
            throw new IOException("Usage: EditDistanceDP fileName");

        String fileName = args[0];
        FileReader reader = new FileReader(fileName);

        FastA fastA = new FastA();
        fastA.read(reader);
        reader.close();

        EditDistance editDistanceDP = new EditDistance();

        if (fastA.size() == 2) {
            int editDistance = editDistanceDP.align(fastA.getSequence(0), fastA.getSequence(1));

            System.out.println("Edit distance is=" + editDistance);

            System.out.println("An optimal alignment=");
            editDistanceDP.traceBackAndShowAlignment(fastA.getSequence(0), fastA.getSequence(1));
        }
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_172\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2018.1.1\lib\idea_rt.jar=55007:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2018.1.1\bin" -Dfile.encoding=UTF-8 -
classpath "C:\Program Files\Java\jdk1.8.0_172\jre\lib\charsets.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\deploy.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\access-bridge-64.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\cldrdata.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\dnsns.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\jaccess.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\jfxrt.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\localedata.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\nashorn.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunec.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunjce_provider.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunmscapi.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunpkcs11.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\zipfs.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\javaws.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jce.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jfr.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jfxswt.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jsse.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\management-agent.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\plugin.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\resources.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\rt.jar;C:\Users\isoch\IdeaProjects\A02_GdB\out\production\A0
2_GdB" A02_GdB.EditDistance C:\Users\isoch\Desktop\sequences2.fasta
Anastasia Grekova, Huajie Chen
Edit distance is=4
An optimal alignment=
CGTCACAAT--TCGTGA

Process finished with exit code 0
```



```
        System.out.println(stack);
    }

    /**
     * main program: reads two sequences in fastA format and computes their optimal alignment score.
     *
     * @param args commandline arguments
     */
    public static void main(String[] args) throws IOException {
        System.out.println("Anastasia Grekova, Huajie Chen");

        if (args.length != 1)
            throw new IOException("Usage: EditDistanceDP fileName");

        String fileName = args[0];
        FileReader reader = new FileReader(fileName);

        FastA fastA = new FastA();
        fastA.read(reader);
        reader.close();

        EditDistance editDistanceDP = new EditDistance();

        if (fastA.size() == 2) {
            int editDistance = editDistanceDP.align(fastA.getSequence( i: 0), fastA.getSequence( i: 1));

            System.out.println("Edit distance is=" + editDistance);

            System.out.println("An optimal alignment=");
            editDistanceDP.traceBackAndShowAlignment(fastA.getSequence( i: 0), fastA.getSequence( i: 1));
```

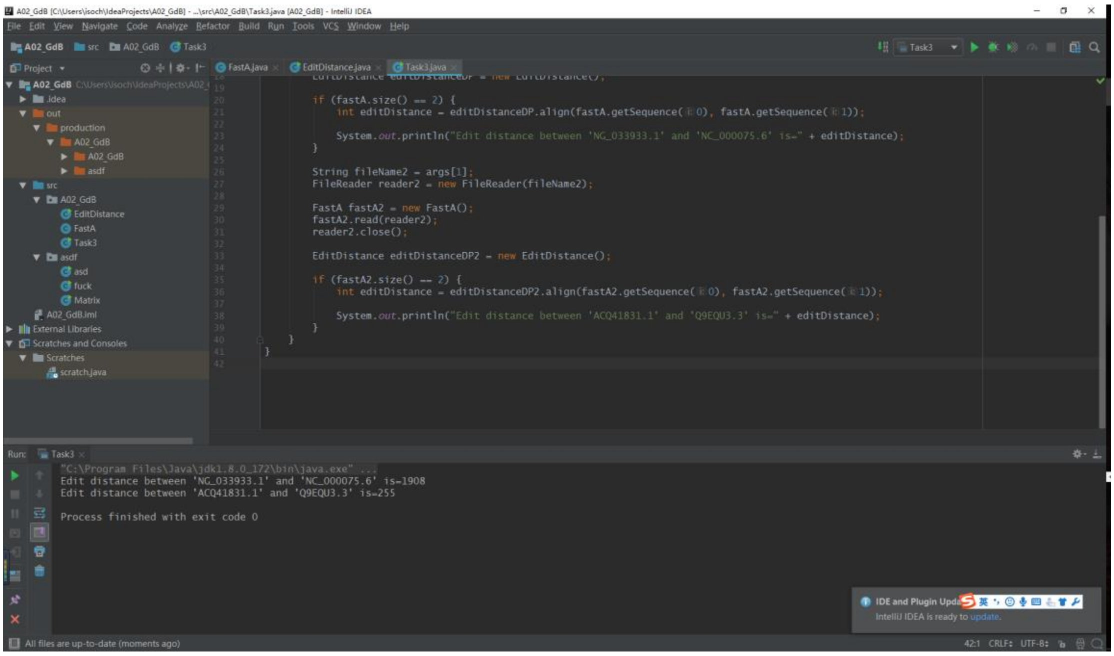
```
"C:\Program Files\Java\jdk1.8.0_172\bin\java.exe" ...
Anastasia Grekova, Huajie Chen
Edit distance is=4
An optimal alignment=
CGTCACAAT--TCGTGA

Process finished with exit code 0
```

Typo: In word 'Huajie'

```
package A02_GdB;

import java.io.FileReader;
import java.io.IOException;

public class Task3 {
    public static void main(String[] args) throws IOException{
        if (args.length != 2)
            throw new IOException("Usage: EditDistanceDP fileName");
        String fileName1 = args[0];
        FileReader reader1 = new FileReader(fileName1);

        FastA fastA = new FastA();
        fastA.read(reader1);
        reader1.close();

        EditDistance editDistanceDP = new EditDistance();

        if (fastA.size() == 2) {
            int editDistance = editDistanceDP.align(fastA.getSequence(0),
fastA.getSequence(1));

            System.out.println("Edit distance between 'NG_033933.1' and 'NC_000075.6'
is=" + editDistance);
        }

        String fileName2 = args[1];
        FileReader reader2 = new FileReader(fileName2);

        FastA fastA2 = new FastA();
        fastA2.read(reader2);
        reader2.close();

        EditDistance editDistanceDP2 = new EditDistance();

        if (fastA2.size() == 2) {
            int editDistance = editDistanceDP2.align(fastA2.getSequence(0),
fastA2.getSequence(1));

            System.out.println("Edit distance between 'ACQ41831.1' and 'Q9EQU3.3' is=" +
editDistance);
        }
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_172\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2018.1.1\lib\idea_rt.jar=50291:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2018.1.1\bin" -Dfile.encoding=UTF-8 -
classpath "C:\Program Files\Java\jdk1.8.0_172\jre\lib\charsets.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\deploy.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\access-bridge-64.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\cldrdata.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\dnsns.jar;C:\Program
```

```
Files\Java\jdk1.8.0_172\jre\lib\ext\jaccess.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\jfxrt.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\localedata.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\nashorn.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunec.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunjce_provider.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunmscapi.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\sunpkcs11.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\ext\zipfs.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\javaws.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jce.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jfr.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jfxswt.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\jsse.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\management-agent.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\plugin.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\resources.jar;C:\Program
Files\Java\jdk1.8.0_172\jre\lib\rt.jar;C:\Users\isoch\IdeaProjects\A02_GdB\out\production\A0
2_GdB"         A02_GdB.Task3         C:\Users\isoch\Desktop\Seq\TLR9_gene.fasta
C:\Users\isoch\Desktop\Seq\TLR9_protein.fasta
Edit distance between 'NG_033933.1' and 'NC_000075.6' is=1908
Edit distance between 'ACQ41831.1' and 'Q9EQU3.3' is=255

Process finished with exit code 0
```


A02_GdB [C:\Users\isoch\IdeaProjects\A02_GdB] - ...\src\A02_GdB\Task3.java [A02_GdB] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
A02_GdB src A02_GdB Task3
Project
FastA.java
EditDistance.java
Task3.java
A02_GdB C:\Users\isoch\IdeaProjects\A02_
.idea
out
production
A02_GdB
A02_GdB
asdf
src
A02_GdB
EditDistance
FastA
Task3
asdf
asd
fuck
Matrix
A02_GdB.iml
External Libraries
Scratches and Consoles
Scratches
scratch.java
Run:
Task3
IDE and Plugin Upda
IntelliJ IDEA is ready to update.
All files are up-to-date (moments ago)
42:1 CRLF: UTF-8:

```java
        EditDistance editDistanceDP = new EditDistance();

        if (fastA.size() == 2) {
            int editDistance = editDistanceDP.align(fastA.getSequence(0), fastA.getSequence(1));

            System.out.println("Edit distance between 'NG_033933.1' and 'NC_000075.6' is=" + editDistance);
        }

        String fileName2 = args[1];
        FileReader reader2 = new FileReader(fileName2);

        FastA fastA2 = new FastA();
        fastA2.read(reader2);
        reader2.close();

        EditDistance editDistanceDP2 = new EditDistance();

        if (fastA2.size() == 2) {
            int editDistance = editDistanceDP2.align(fastA2.getSequence(0), fastA2.getSequence(1));

            System.out.println("Edit distance between 'ACQ41831.1' and 'Q9EQU3.3' is=" + editDistance);
        }
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_172\bin\java.exe" ...
Edit distance between 'NG_033933.1' and 'NC_000075.6' is=1908
Edit distance between 'ACQ41831.1' and 'Q9EQU3.3' is=255

Process finished with exit code 0
```

1)

Edit distance between 'NG_033933.1' and 'NC_000075.6' is=1908

Edit distance between 'ACQ41831.1' and 'Q9EQU3.3' is=255

2)

Firstly, 3 bases form 1 single codon corresponding to one specific amino acid, which denotes that the translated amino acid sequence is meant to be at least 3 times shorter than the original DNA sequence. Secondly, different sequences of DNA might lead to the same translated AA sequence (i.e. “silent mutation, in which the sequence is change but the translation result remains the same) because of different codons matching the same AA. Thirdly, there are non-coding regions and introns in the gene and they are not to be translated in the end. If the non-coding region is long enough, the edit distance is to be reduced significantly. Fourthly, the alternative splicing effect can also contribute to the increasement of edit distance (depending on evolutionary conditions) .

3)

The secondary structure or above is not considered in this situation. Therefore, the only thing that we need to take into consideration is the AA sequence. As far as we are concerned, the features of AA will not affect the condensation of -COOH and -NH2. So, we tend to believe that in this case the features of AA will not affect the edit distance.

4)

Edit distance is mainly used in comparing the DNA sequences. As for AA sequences, another method called “Similarity Score” is better. The similarity in this case should be maximized. In case of proteins we are also interested in some biochemical properties of amino acids, that can lead to insignificant changes. For example, it doesn’t change much, when one small nonpolar amino acid has been changed to the other (like isoleucine -> leucine). That is why it is better to maximize score of similarities than minimize score of differences by proteins. In that case its useful to implement Needlemann-Wunsch (Sequence Similarity).