

## PROGRAMMING ASSIGNMENT 3

**Course Instructors:** Assoc. Prof. Dr. Erkut Erdem, Assoc. Prof. Dr. Hacer Yalın Keleş, Asst. Prof. Dr. Adnan Özsoy

**TAs:** Sümeyye Meryem Taşyürek, Hikmet Can Doğanç

**Programming Language:** Java (OpenJDK 11)

**Due Date:** Friday, 02.05.2025 (23:59:59)

### 1 FloraX

After traveling to another planet for an expedition, humanity encountered something totally unexpected. There was a type of alien flora that couldn't be seen by any of our cameras. It was like the plants were playing hide and seek with our technology — completely invisible, but definitely there.

Once this discovery was made, the team decided to return to the space station to rethink the mission. The situation was way more complex than anyone imagined. So, a new plan was formed: assemble a brave team of scientists to go back and study this mysterious flora in detail. Naturally, a call went out, and a ton of people applied — everyone wanted a chance to be part of history.

And here's the cool part — you were selected as the leader of this new team.

Your mission? Lead your team into the unknown, study this alien flora, and figure out if the planet is safe enough for humanity to start a new life there. Is it dangerous? Friendly? Weirdly sentient? Time to find out. The future could depend on what you discover.



## 2 Part 1: Identify Genomes

**You will parse XML and use graphs to create your structure.**

With the discovery of Alien Flora, scientist had their work cut for themselves. Now they have a whole world of flora to classify and research. For this goal, they started taking genome samples from flora, and started classifying them. In order to classify, scientists need to find genome groups by creating a genome graph.

For this part, read every genome, their links, and their evolutionFactor. Create a GenomeCluster object (Graph) of Genome objects. An example XML part is provided below for a clear understanding.

```
<data>
  <genomes>
    <genome>
      <id>G102</id>
      <evolutionFactor>12</evolutionFactor>
      <links>
        <link>
          <target>G103</target>
          <adaptationFactor>5</adaptationFactor>
        </link>
        <link>
          <target>G104</target>
          <adaptationFactor>10</adaptationFactor>
        </link>
      </links>
    </genome>
    <genome>
      <id>G103</id>
      <evolutionFactor>15</evolutionFactor>
      <links>
        <link>
          <target>G104</target>
          <adaptationFactor>2</adaptationFactor>
        </link>
      </links>
    </genome>
    <genome>
      <id>G104</id>
      <evolutionFactor>10</evolutionFactor>
      <links/>
    </genome>
    <genome>
      <id>G201</id>
      <evolutionFactor>9</evolutionFactor>
      <links>
        <link>
          <target>G202</target>
          <adaptationFactor>4</adaptationFactor>
        </link>
      </links>
    </genome>
    <genome>
      <id>G202</id>
```

```
<evolutionFactor>11</evolutionFactor>
  <links/>
</genome>
</genomes>
...
</data>
```

Listing 1: Sample XML Input For Genomes

You will read this XML file and create graphs according to it. In this example we can see 2 Genome Clusters. One being [G102, G103, G104] and the other one is [G201, G202]. For this XML part your output should be:

```
##Start Reading Flora Genomes##
Number of Genome Clusters: 2
For the Genomes:  [[G103, G102, G104], [G201, G202]]
##Reading Flora Genomes Completed##
```

### 3 Part 2: Find the Minimum Energy to Evolve

**You will use minimum spanning tree to find the minimum evolution factor between Genome Clusters.**

After searching through the flora, scientist discovered major changes in it. Upon detailed research, scientists found that the flora has major differences in their genomes. This showed that some older flora has the capability to evolve very fast. As this is a breakthrough in science, scientists created an algorithm to find the ancient flora that caused all of these evolutions.

For this part, read every possibleEvolutionPair. Do the following for each element in pairs, find its cluster, find genome with minimum evolutionFactor in the respective cluster. Then use equation 1. An example XML part is provided below for a clear understanding.

$$Answer = (Pair[1]_{minEvolutionFactor} + Pair[2]_{minEvolutionFactor})/2 \quad (1)$$

An example XML part is provided below for a clear understanding.

```
<data>
...
<possibleEvolutionPairs>
  <pair>
    <firstId>G102</firstId>
    <secondId>G202</secondId>
  </pair>
  <pair>
    <firstId>G201</firstId>
    <secondId>G202</secondId>
  </pair>
</possibleEvolutionPairs>
...
</data>
```

---

Listing 2: Sample XML Input For Evolution Pairs

For finding the minimum energy to evolve, we will go through every pair in possible evolution pairs. For first pair we will choose the cluster that has these id's, and find the elements with lowest evolution factor. We will use equation 1.

Given:

- For G102, lowest evolution factor in same cluster is 10 with G104.
- For G202, lowest evolution factor in same cluster is 9 with G201.

$$Answer = (Pair[1]_{minEvolutionFactor} + Pair[2]_{minEvolutionFactor})/2 \quad (2)$$

$$= (G104_{minEvolutionFactor} + G201_{minEvolutionFactor})/2 = 19/2 = 9.5 \quad (3)$$

For second pair, because elements are on the same cluster, we will return -1.

```
##Start Evaluating Possible Evolutions##  
Number of Possible Evolutions:  2  
Number of Certified Evolution:  1  
Evolution Factor for Each Evolution Pair:  [9.5, -1]  
##Evaluated Possible Evolutions##
```

## 4 Part 3: Find Possible Adaptations

**You will use Dijkstra's algorithm to find the minimum adaptability factor in a Genome Cluster.**

After analyzing the genomes, scientist also discovered major adaptations to the environment that can be confused with evolutions. To understand whether a change is adaptation or evolution, scientists created an algorithm to check genomes. If the genomes are in the same cluster, they are adaptations; on the other hand, if they are not in the same cluster, they are evolutions.

For this part, read every possibleAdaptationPair. Check if two genomes are in same cluster, if they are, find minimum path by using Dijkstra's algorithm and return the minimum path, if they are not on same cluster, just return -1.

An example XML part is provided below for a clear understanding.

```
<data>  
...  
<possibleAdaptationPairs>  
  <pair>  
    <firstId>G102</firstId>  
    <secondId>G202</secondId>  
  </pair>  
  <pair>  
    <firstId>G102</firstId>  
    <secondId>G104</secondId>
```

```
</pair>
</possibleAdaptationPairs>
</data>
```

Listing 3: Sample XML Input

For finding the minimum energy to adapt, we will go through every pair in possible adaptation pairs. For first pair we will return -1 as they are on different clusters. For second pair, because the elements are on same cluster we will check for shortest genetic path. This pair has 2 possible paths: [G102 - G103 - G104] and [G102 - G104]. First path has adaptationFactor of 7. While second has 10. So you should return 7.

For this XML part your output should be:

```
##Start Evaluating Possible Adaptations##
Number of Possible Adaptations:  2
Number of Certified Adaptations:  1
Adaptation Factor for Each Adaptation Pair:  [-1, 7]
##Evaluated Possible Evolutions##
```

## 5 Complete Example

```
<data>
  <genomes>
    <genome>
      <id>G102</id>
      <evolutionFactor>12</evolutionFactor>
      <links>
        <link>
          <target>G103</target>
          <adaptationFactor>5</adaptationFactor>
        </link>
        <link>
          <target>G104</target>
          <adaptationFactor>10</adaptationFactor>
        </link>
      </links>
    </genome>
    <genome>
      <id>G103</id>
      <evolutionFactor>15</evolutionFactor>
      <links>
        <link>
          <target>G104</target>
          <adaptationFactor>2</adaptationFactor>
        </link>
      </links>
    </genome>
    <genome>
      <id>G104</id>
      <evolutionFactor>10</evolutionFactor>
      <links/>
    </genome>
  </genomes>
</data>
```

```
<genome>
  <id>G201</id>
  <evolutionFactor>9</evolutionFactor>
  <links>
    <link>
      <target>G202</target>
      <adaptationFactor>4</adaptationFactor>
    </link>
  </links>
</genome>
<genome>
  <id>G202</id>
  <evolutionFactor>11</evolutionFactor>
  <links/>
</genome>
</genomes>

<possibleEvolutionPairs>
  <pair>
    <firstId>G102</firstId>
    <secondId>G202</secondId>
  </pair>
  <pair>
    <firstId>G201</firstId>
    <secondId>G202</secondId>
  </pair>
</possibleEvolutionPairs>

<possibleAdaptationPairs>
  <pair>
    <firstId>G102</firstId>
    <secondId>G202</secondId>
  </pair>
  <pair>
    <firstId>G102</firstId>
    <secondId>G104</secondId>
  </pair>
</possibleAdaptationPairs>
</data>
```

Listing 4: Sample XML Input

The console output for the example XML should be:

```
##Start Reading Flora Genomes##
Number of Genome Clusters: 2
For the Genomes:  [[G103, G102, G104], [G201, G202]]
##Reading Flora Genomes Completed##
##Start Evaluating Possible Evolutions##
Number of Possible Evolutions: 2
Number of Certified Evolution: 1
Evolution Factor for Each Evolution Pair:  [9.5, -1]
##Evaluated Possible Evolutions##
##Start Evaluating Possible Adaptations##
Number of Possible Adaptations: 2
Number of Certified Adaptations: 1
Adaptation Factor for Each Adaptation Pair:  [-1, 7]
##Evaluated Possible Evolutions##
```

## 6 Starter Code

You must use the [starter code](#) provided here.

## 7 Code Run Configurations

Your code will be compiled and run as follows:

```
javac *.java -d .
java Main <AlienFlora>
```

## 8 Grading Policy

- Evolution Tests - 45%
- Adaptation Tests - 45%
- Output Tests - 10%

## 9 Important Notes

- **Usage of any external libraries is forbidden.**
- Use the provided [starter code](#).
- Do not miss the deadline: **Friday, 02.05.2025 (23:59:59)**.
- Save all your work until the assignment is graded.
- The assignment solution you submit must be your original, individual work. Duplicate or similar assignments are both going to be considered as cheating.

- You can ask your questions via Piazza (<https://piazza.com/hacettepe.edu.tr/spring2025/bbm204>), and you are supposed to be aware of everything discussed on Piazza.
- You will submit your work via <https://submit.cs.hacettepe.edu.tr/> with the file hierarchy given below:

```
b<studentID>.zip
├── Main.java <FILE>
├── Genome.java <FILE>
├── GenomeCluster.java <FILE>
└── AlienFlora.java <FILE>
```

- The name of the main class that contains the main method should be **Main.java**. The main class and all other classes should be placed directly (no subfolders) into a zip file named **b<studentID>.zip**.
- Feel free to create subdirectories if necessary, but they should also be in **b<studentID>.zip** directory.
- This file hierarchy must be zipped before submitted (not .rar, only .zip files are supported).

## 10 Academic Integrity Policy

All work on assignments **must be done individually**. You are encouraged to discuss the given assignments with your classmates, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in pseudocode) **will not be tolerated**. In short, turning in someone else's work (including work available on the internet), in whole or in part, as your own will be considered as **a violation of academic integrity**. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.



**The submissions will be subjected to a similarity check. Any submissions that fail the similarity check will not be graded and will be reported to the ethics committee as a case of academic integrity violation, which may result in suspension of the involved students.**