

WINTER UNIVERSITY 2022

JAVA EXERCISE

Brotherly Feud

Semi Nario set off on a quest to win heart of princess Peach. He has already defeated 4 dragons, 2 quality assurance engineers and bamboozled dozen witches. The only obstacle that lays between him and princess Peach is abyss deeper than your eight grade essays. The only safe way through the abyss is array of platforms. Nario has decided to jump from one platform to another, to reach to the other side of the abyss. Just before Nario is ready to jump on the first platform, he notices his older brother Super Mario. As their eyes lock, they both realize that they are after the very same woman. Contrary to popular belief Super Mario is quite rude and obnoxious man. Decades of fame and mushrooms have poisoned Mario's mind. Help Semi Nario to reach princess Peach before his annoying big brother Super Mario.

Setup

If on windows execute "gradlew.bat build" in project root folder. If on unix or linux execute "gradlew build". You can also use your own locally installed gradle, but it might result in mismatching gradle versions.

Alternatively, you can skip these previous steps by installing smart IDE like IntelliJ which does forementioned automatically.

Assignment

Read game track from csv files and complete the track with **least amount of jumps possible**.

Game Rules

- Track consists of platforms which are stored in csv files (see src/main/resources/platforms.csv).
- First line of the csv file is always header row. It does not contain actual platforms.
- Each line contains one platform. First platform is always indexed as 0 and platform indexes grow by one each row.
- You start the game with 500 points.
- You start the game from platform with index 0. This platform is unlocked by default.
- To jump to the next platform, you must unlock it first. Unlock cost is stored in csv file in the **cost** column.
- You can unlock the platform by jumping to it while having sufficient points.
- Platform cost is in range [0, 20000]
- You can earn points by jumping back to previously unlocked platforms.
- Game is completed when you've reached the to the last platform.

Example Game

Imagine following csv file:

```
index, cost
0, 100
1, 200
2, 400
3, 100
```

You start the game from platform 0 with 500 points.

- You jump to platform 1. It is the first jump to that platform, so you're going to have to pay the cost. **Points: $500 - 200 = 300$**
- You cannot jump to platform 2, because it costs 400 points, but you only have 300. Time to go back. You jump to platform 0. **Points: $300 + 100 = 400$.**
- You jump to platform 1. **Points: $400 + 200 = 600$**
- After taking detour you finally have enough points to jump to platform 2, so you jump to there. **Points: $600 - 400 = 200$**
- You jump to platform 3. **Points: $200 - 100 = 100$**

Congratulations, game completed with 5 jumps.

Additional Important Instructions

TODOs have been left in the Game.java file. Implementation is up to you, just **make sure that whenever you jump to platform you also invoke `jumpTo(Platform platform)` method**. You don't need to embed your program logic in that method. Our Nortal submission review heroes need that method to validate your solution.