

Assignment #5 – Greedy Method

Total points: 60

This assignment will use greedy method to solve two problems: fractional knapsack problem and task scheduling problems.

Objectives:

The main objectives of this assignment are:

- Implement Fractional Knapsack problem with the help of PQ
- Implement Task Scheduling problem with the help of PQ
- Write test cases

The Assignment:

Resource that you could reference:

- Class notes
- HW4
- Greedy slides page 13-15

Your Job:

1. Download and import the code starter in to Eclipse
2. Write test cases for method MaximumValue() in FractionalKnapsackTest.java **before you implement the method.**
3. Implement method MaximumValue() in FractionalKnapsack.java. You may copy necessary files from previous programming assignments.
4. Write test cases for method NumOfMachines() in TaskSchedulingTest.java **before you implement the method.**
5. Implement method NumOfMachines() in TaskScheduling.java. You may copy necessary files from previous programming assignments.

Allowed Imports:

```
import net.datastructures.*;  
import java.util.Comparator;
```

Submission:

First, look at the following checklist:

1. Do you ever `import` any class besides the ones that is allowed?
2. Did you test your code?
3. Is the indentation easily readable? You can have Eclipse correct indentation by highlighting all code and select "Source → Correct Indentation".
4. Are comments well organized and concise?

If you have reviewed your code and all the criteria on the checklist are acceptable, follow the submission procedure to export **prog5.zip**.

Grading Criteria (total: 60p)

- Implement `MaximumValue` correctly: 20
- Implement `NumOfMachines` correctly: 25
- Test cases for `MaximumValue`: 5
- Test cases for `NumOfMachines`: 5
- Programming Style and Comments: 5

FAQ

Q: How to get decimal result for division?

A: Cast one of the operands to be double.

```
int x=1, y=2;
double z;
z = (double) x /y
```

Q: How to compare doubles in Junit?

A: Use a small delta value, such as 0.000001.

[https://junit.org/junit4/javadoc/4.8/org/junit/Assert.html#assertEquals\(double,%20double,%20double\)](https://junit.org/junit4/javadoc/4.8/org/junit/Assert.html#assertEquals(double,%20double,%20double))

assertEquals

```
public static void assertEquals(double expected,
                                double actual,
                                double delta)
```

Asserts that two doubles or floats are equal to within a positive delta. If they are not, an [AssertionError](#) is thrown. If the expected value is infinity then the delta value is ignored. NaNs are considered equal: `assertEquals(Double.NaN, Double.NaN, *)` passes

Parameters:

`expected` - expected value

`actual` - the value to check against `expected`

`delta` - the maximum delta between `expected` and `actual` for which both numbers are still considered equal.