Programming Assignment #3

Programming assignments are to be done individually. You may discuss the problem and general concepts with other students, but there should be no sharing of code. You may not submit code other than that which you write yourself or is provided with the assignment. This restriction specifically prohibits downloading code from the Internet. If any code you submit is in violation of this policy, you will receive no credit for the entire assignment.

The goals of the project are to ensure that you:

- Understand dynamic programming.
- Are able to argue for optimal substructure of the problem and define recurrence for optimal solution.
- Are able to implement and test the algorithm for the optimal solution proposed.

The Problem

It's the end of the semester, and you're taking n classes. Each class will be graded on a scale of 0 to 10, where a higher number is a better grade. Your goal is to maximize your average grade on the n classes. (Note: this is equivalent to maximizing the *total* of all grades, since the difference between the total and the average is just the factor n.)

You have a finite number of hours $H > n$ to complete all of your coursework; you need to decide how to divide your time. H is a positive integer, and you can spend an integer number of hours on each class. Assume your grades are deterministically based on time spent; you have a set of functions $\{f_i : 0, 1, \ldots, n-1\}$ for your n courses, and if you spend $h \leq H$ hours on course i, you will get a grade of $f_i(h)$. The functions f_i are nondecreasing; spending more time on a class will not *lower* your grade in the course. On each class, you can spend from a minimum of 0 to a maximum of 9 hours. (For information how these functions are given to you, please see the Test Cases Sample Input).

Given these functions, decide how many hours to spend on each class (again, in integer numbers of hours) to maximize your average grade. Your algorithm should be polynomial in n and H.

How to store your solution

You will store your results in array named **studentoutput**. It is an array of objects of result type (given in the starter code). Each object has a parameter named **hour** which represents how many hours you need to study for a class, and a parameter named **grade** which represents expected grade in that class if you study for the number of hours defined in parameter **hour**. The index of this array represents class number. Suppose you have studentoutput[0].hour=9 and studentoutput[0].grade=8. It means you will study for 9 hours for class 0 and will achieve a grade of 8 in class 0. Further detailed explanation of how your final array will look like is given in following sections of this document.

You also need to write a brief report in which you should argue that the problem has optimal substructure, define the value of the optimal solution (i.e., by providing the recursive definition), describe an algorithm for iteratively computing the value of the optimal solution, and describe an algorithm for recreating the optimal solution (i.e., mapping your H hours to your n projects). Your implementation should provide a Java implementation of your algorithms. **There is a commented section in Assignment3.java inside which you need to write this brief report.**

Explanation of Test Cases and Expected Output:
TEST CASE 1
Input File will have following values
10 20
0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 8 10
0 2 4 6 8 8 8 8 8 8
1 1 1 1 5 5 5 7 7 7
4 4 4 4 5 5 8 9 10 10
3 3 3 3 3 3 8 8 8 10
1 2 3 4 5 6 7 8 9 10
1 3 5 7 8 9 10 10 10 10
2 3 4 5 5 5 6 6 6 6
3 4 5 6 7 8 9 9 9 9

The first line specifies the two parameters of the problem: The first value 10 means that total number of classes is 10. The second value means that you have 20 hours in total to study. After this, each line corresponds to a class and describes the grade you will receive as a function of the number of hours you spend on that class (i.e., this is how the functions f_i are encoded). Let us take as example the second line (after 10 20). It means that if you study for 0 hours you will get 0 grade in class 0. If you study for 1 hour, you will get grade 1. Similarly, if you study for 9 hours you will get grade 9 in class 0. To explain it further, consider last line in the example. It means that if you study for 0 hours, you will get grade 3 in class 9. If you study 6 hours, you will get grade 9 and if you study for 9 hours, you will still get grade 9 in class 9.

The optimal solution for above problem:
You study for 9 hours for class 0 to get grade of 9
You study for 4 hours for class 1 to get grade of 5
You study for 4 hours for class 2 to get grade of 8
You study for 0 hours for class 3 to get grade of 1
You study for 0 hours for class 4 to get grade of 4
You study for 0 hours for class 5 to get grade of 3
You study for 0 hours for class 6 to get grade of 1
You study for 3 hours for class 7 to get grade of 7
You study for 0 hours for class 8 to get grade of 2
You study for 0 hours for class 9 to get grade of 3

You will store it as follows.
studentoutput[0].setHour(9)
studentoutput[0].setGrade(9)
studentoutput[1].setHour(4)
studentoutput[1].setGrade(5)
studentoutput[2].setHour(4)
studentoutput[2].setGrade(8)
studentoutput[3].setHour(0)
studentoutput[3].setGrade(1)
studentoutput[4].setHour(0)

```
studentoutput[4].setGrade(4)
studentoutput[5].setHour(0)
studentoutput[5].setGrade(3)
studentoutput[6].setHour(0)
studentoutput[6].setGrade(1)
studentoutput[7].setHour(3)
studentoutput[7].setGrade(7)
studentoutput[8].setHour(0)
studentoutput[8].setGrade(2)
studentoutput[9].setHour(0)
studentoutput[9].setGrade(3)
```

TEST CASE 2

Input File will have following values
10 20
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
It means that there are 10 total classes and you again have 20 hours to study. If you study for 0 hours in each class, you will earn grade 1 in each class. If you study for 9 hours for a class, then you will earn grade 10 in that class.
The optimal solution for above problem:
You study for 9 hours for class 0 and get grade of 10.
You study for 9 hours for class 1 and get grade of 10.
You study for 2 hours for class 2 and get grade of 3.
You study for 0 hours for rest of classes and get grade of 1 in each of them.
Note: This test case has ties and you should resolve ties by choosing classes with lower ids: class 0 is preferred to class 1, class 1 is preferred to class 2, and so on.

You will store it as follows.
studentoutput[0].setHour(9)
studentoutput[0].setGrade(10)
studentoutput[1].setHour(9)
studentoutput[1].setGrade(10)
studentoutput[2].setHour(2)
studentoutput[2].setGrade(3)
studentoutput[3].setHour(0)
studentoutput[3].setGrade(1)
studentoutput[4].setHour(0)
studentoutput[4].setGrade(1)
studentoutput[5].setHour(0)
studentoutput[5].setGrade(1)
studentoutput[6].setHour(0)
studentoutput[6].setGrade(1)
studentoutput[7].setHour(0)
studentoutput[7].setGrade(1)
studentoutput[8].setHour(0)
studentoutput[8].setGrade(1)
studentoutput[9].setHour(0)
studentoutput[9].setGrade(1)

Provided Code

You are provided very little starter code in this assignment. You are given the following:

- `TestAssignment3.java`: This class is used to test your code which you will implement in Assignment3.java file.
- `Assignment3.java`: It has a function **public result[] compute(int totalHours, result[] studentoutput)** inside which you will implement your code. You will return the array named **studentoutput** which should hold the values you computed (your solution). As described before, studentoutput is an array of objects of type result. Each object has two parameters, namely hour and grade and index of this array represents class number.
  We will call this function during grading. We will supply total number of hours you have for study as first parameter of this function and an empty studentoutput array(all values initialized to 0). The array size is already defined for you in TestAssignment3.java. Moreover, Assignment3.java has three variables **totalClasses**, **maxGrade** (10*totalClasses because maximum in each class can be 10) and a 2D array named **gradearray**. These 3 are already initialized for you using function named **initialize**. Row number of gradearray represents class number and column number is used to represent number of hours you study for that class where as value of cell represents grade. For testcase 1, gradearray look like this:

$$gradearray = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 8 & 10 \\ 0 & 2 & 4 & 6 & 8 & 8 & 8 & 8 & 8 & 8 \\ 1 & 1 & 1 & 1 & 5 & 5 & 5 & 7 & 7 & 7 \\ 4 & 4 & 4 & 4 & 5 & 5 & 8 & 9 & 10 & 10 \\ 3 & 3 & 3 & 3 & 3 & 3 & 8 & 8 & 8 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 3 & 5 & 7 & 8 & 9 & 10 & 10 & 10 & 10 \\ 2 & 3 & 4 & 5 & 5 & 5 & 6 & 6 & 6 & 6 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 9 & 9 & 9 \end{bmatrix}$$

You may not change or augment the provided code in any way. Your solution should be provided in `Assignment3.java`.

What to Submit

Failure to follow these instructions **will** result in a deduction of points.

(a) Implement your solution in `Assignment3.java`. Document your code (with comments and self-documenting code). Include your brief report in the comments.

(b) Copying of code is strictly prohibited. Copying code will result in 0 points in Assignment 3.

(c) Please fill in provided header information of Assignment3.java.

(d) Only your most recent submission to Canvas will be graded. You can submit as many times as you like, and we will only grade the most recent submission.

(e) **DO NOT USE PACKAGE STATEMENTS.** Your code will fail to compile in our grader if you do and it won't be regarded if your program fails to compile due to package imports.

Submission

Submitting any file other than Assignment3.java will result in a penalty of up to 25 points.