

EE 360C

Xiangxing Liu

xl5587

Professor Touba

TA: Mona Mishra

Lab 3 Report

1. Problem Definition:

a) Optimal Substructure

In order to show that the problem has optimal substructure, we have to define the sub-problems of this problem. The sub-problem is to compute the maximum grade of each project/class at each unite time. So, for each new project, we have to compute the maximum total grade of current project, and then compare the result to the maximum total grade of the previous project (previous sub-problem). After all these comparisons, we will reach the last available project/class, and we will be able to retain the correct maximum grade for this lab.

b) Recursive Definition:

We have to build two matrices, one matrix $grades[i][h]$ to record the max grade for each class/project ID i for each hour h .

$$grades[i][H] = \max\{\max(0 < h \leq \text{total hours } H)\{grades[H - h][i - 1] + fi(h)\}, grades[i - 1][H]\}$$

We also have the other matrix $hours[i][h]$ to keep track of hours used.

$$hours[i][H] = \begin{cases} -1 & \text{if } grades[i][H] = grades[i - 1][H] \\ \text{hours that maximized } grades[i][H] & \text{otherwise} \end{cases}$$

2. Pseudo – Code

DynamicSolution(i, h)

Initialize 2D array $grades[i][h]$ and $hours[i][h]$, where i = number of classes, and h = hours

If there are zero classes, or zero total hours left, then

Return

For each class i from 1 to n

For each hour h from 0 to H , where H = total hours allowed

max = 0

hour = 0

For each hour j from 0 to h

If grade of this j hour + $\text{grades}[i - 1][h - j] > \text{max}$, then

max = grade of this hour

hour = j

If $\text{max} > \text{grades}[i - 1][h]$, then

$\text{grades}[i][h] = \text{max}$

$\text{hours}[i][h] = \text{hour}$

else

$\text{grades}[i][h] = \text{grades}[i - 1][h]$

$\text{hours}[i][h] = -1$

3. Test Report

For test cases provided, test results were verified manually. And correct results were shown on the console as the output instructions provided.

One thing that interested me was that all grade functions provided return a grade that is progressively related to the amount of input hours. As a result, the maximum grade always shown with maximum hours. I wanted to test if my algorithm would work if grades returned by the grade function are regressively related to input hours. Thus, I designed my own grade function to return a lower grade as the number of hours increase.

The result of the algorithm using my grade function decided to use minimum number of hours to maximize total grades, which was correct as I intended.