

**Zi Zhou Wang**  
**Zw3948**

### **LAB 3 REPORT**

**Problem Definition** - The overall problem is to maximize the grade of a student by planning their time allotments for projects. Given a set of non-decreasing functions, we must maximize the student's total grade by determining an hour allotment for each function.

**Optimal Substructure** - We can calculate the maximum grade for a certain number of classes by taking the maximum result of the previous class. We have to take into consideration all hours for the current class we are considering and compare these results to the previous result at the current hour we are considering. Maximizing the grade between all of these comparisons will give the maximum grade possible.

**Recursive Definition** - The recursive function  $G[\text{class}, \text{hour}]$  is the highest grade possible at that point. We will also define  $G[\text{class}, \text{hour}]$  equal to 0 when class is equal to zero.

$G[\text{class}, \text{hour}] = \max\{\max\{G[\text{class}-1, \text{hour} - i] + f_{\text{class}}(i)\}, G[\text{class}-1][\text{hour}]\}$  where  $1 \leq i \leq \text{hour}$   
While running this algorithm we have a second matrix which represents the hours used by a particular class. This will keep track of the values throughout its execution in order to reconstruct the solution after finding the maximum score.  $H[\text{class}, \text{hours}] = -1$  if  $G[\text{class}][\text{hour}] < G[\text{class}-1][\text{hour}]$  and  $H[\text{class}, \text{hours}] = \text{the current time}$  if  $G[\text{class}][\text{hour}] > G[\text{class}-1][\text{hour}]$

#### **Iterative Algorithm -**

```
function solve(hours)
  initialize G[H,j]
  if we have solved this problem, return the solution
  for all classes j
    for all hours H
      find an i that maximizes  $G[h-i][j-1] + f_j(i)$ 
       $G[H][j] = \text{the max of the previous maximum and } G[H][j-1]$ 

  calculate solution by finding the best values in G for each j
  save this solution
  return the solution
```

**Testing Solution** - All testing was manually verified by hand. Provided test cases were used, included CustomGrade, SquareGrade, and SquareRootGrade. These functions were used with various values of number of classes, and hours to test for the accuracy of the program.