COMP3121-Ass3-Q4

z5302513, Kihwan Baek

July 2021

1. Assumption

For every $i \leq n$, we assume that opt(i,j) (j = 1 or 2 or 3) is the total sum of enjoyment by day i.

2. Base cases

The base case is opt(1,j) = e(1,j) (j = 1 or 2 or 3) which is the maximum enjoyment of the first day.

3. Subproblem and Recursion

To get opt(i,j), we need to add opt(i-1,k) (k != j, k = 1 or 2 or 3) and e(i,j). So, we can divide the cases as 3 based on activities to get opt(i,j).

1) opt(i,j) = max{opt(i-1,2),opt(i-1,3)} + e(i,1) if j = 1
2) opt(i,j) = max{opt(i-1,1),opt(i-1,3)} + e(i,2) if j = 2
3) opt(i,j) = max{opt(i-1,1),opt(i-1,2)} + e(i,3) if j = 3

3. How to obtain the final solution

Starting from opt(1,j_1), it finishes with opt(n,j_n) (j_n = 1 or 2 or 3). From the three different opt(n,1), opt(n,2) and opt(n,3), the final solution will be the maximum of these values.

4. Time complexity We have 3 different amounts of enjoyment for each day. And also, we need to calculate the sum of enjoyments for n days. So, it takes $O(3) \times O(n) = O(n)$ time.