Q3

You are on vacation for N days at a resort that has three possible activities. For each day, for each activity, you've determined how much enjoyment you will get out of that activity if you do it on that particular day (the same 1 activity might give you a different amount at different days). However, you are not allowed to do the same activity two days in a row. What is the maximum total enjoyment possible? (20 marks)

Solution

Assumption: we are given a 2d array of size Nx3 called enj where N is the number of days and 3 are for the number of activities.

e.g. enj[i][1] -> corresponds to doing activity 1 on day i.

Base case : is day 0 which would return maximum total enjoyment of 0

Subproblem: for every $1 \leq i \leq n$, we find the maximum enjoyment of the 3 activities from the previous day and update the current days enjoyment accordingly. This would mean

- For $enj[i][0]$ we will find the maximum enjoyment between activity 1 and 2 from the previous day (i-1) and add the enjoyment value for activity 0 on day i, to the current total.
    - $enj[i][0] += max\ (enj[i-1][1], enj[i-1][2])$
- For $enj[i][1]$ we will find the maximum enjoyment between activity 0 and 2 from the previous day (i-1) and add the enjoyment value for activity 1 on day i, to the current total.
    - $enj[i][1] += max\ (enj[i-1][0], enj[i-1][2])$
- For $enj[i][2]$ we will find the maximum enjoyment between activity 0 and 1 from the previous day (i-1) and add the enjoyment value for activity 2 on day i, to the current total.
    - $enj[i][2] += max\ (enj[i-1][0], enj[i-1][1])$

Final solution: when i equals n - 1 return the maximum values between the 3 activities. This would be

- $max(enj[n-1][0], max\ (enj[n-1][1], enj[n-1][2]))$

Time complexity : Iterating through the 2d array of length n would take $O(n)$ and all other operations occur in constant time. Thus the total complexity of the algorithm will take $O(n)$.