Q3

Setup

Assume three activities are named 0, 1 and 2. And $E[i][0]$, $E[i][1]$ and $E[i][2]$ are stand for the enjoyment that get out of 0, 1 and 2 on ith day respectively.

Subproblem

For $1 \leq i \leq N$, we define the subproblem $P(i)$ to be "find the maximum total enjoyment from the first day to ith day and the same activity are not allowed between two consecutive days".

Let $T(i)$ be the total duration of the optimal solution to the subproblem $P(i)$.

Build-up order

Solve the subproblems in the order $P(1), \ldots, P(n)$.

Base case

$$T[0][0] = E[0][0]$$

$$T[0][1] = E[0][1]$$

$$T[0][2] = E[0][2]$$

$$T[0] = max\{T[0][0], T[0][1], T[0][2]\}$$

As, we are not allowed to do same activity two days in a row, so we use $T[i][a]$ to store the case that total enjoyment ends with a activity on ith day.

Recursion

For all $i > 1$, we have

$$T[i][0] = max\{T[i-1][1] + E[i][0], T[i-1][2] + E[i][0]\}$$

$$T[i][1] = max\{T[i-1][0] + E[i][1], T[i-1][2] + E[i][1]\}$$

$$T[i][2] = max\{T[i-1][0] + E[i][2], T[i-1][1] + E[i][2]\}$$

$$T[i] = max\{T[i][0], T[i][1], T[i][2]\}$$

Final solution

The final solution is given by,

$$T[n] = max\{T[n][0], T[n][1], T[n][2]\}$$

Time complexity

Clearly such algorithm runs in time linear in the number of all N days.