**Problem :**

[**Maximum subsequence sum such that no three are consecutive 2 Moderate 15 mins 161**](https://www.codingninjas.com/codestudio/problems/maximum-subsequence-sum-such-that-no-three-are-consecutive_1215007)

**Approach1 :** Using Top Down , 2-D DP , time = O(n\*3) = O(n),space=O(n)

-> Take or not take approach,

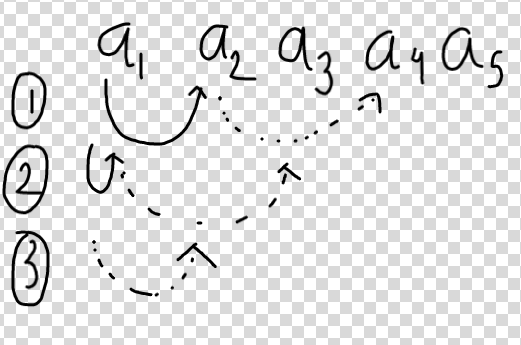
If you decide to take current element in the subsequence and ‘k’ consecutive elements are visited already,pass k+1 to next number showing that k+1 consecutive elements are already taken, so if 2 consec. are already taken , the current element can’t be taken,and need to pass 0 to next number as the current number wasn’t taken

-> If you decide to not take the number , pass 0 to next element showing that no consecutive elements were taken.

**Code using 2-D DP :** [**https://ideone.com/B8YGFz**](https://ideone.com/B8YGFz)

**Approach 2 :** Using 1-D DP, time = O(n),space =O(n),

(here space can be optimized to O(1),if done iteratively,traversing from backward,as we can store values in same array which is given in question.)



-> If you think carefully, then from some index only 3 situations are possible,

1. Take a[i],a[i+1] and then recursively call index ‘i+3’ for maximum sum from there.

So this case allows us to include a[i] & a[i+1] in the subsequence .

1. Take a[i] and recursively call ‘i+2’ to find max sum from there,this case allows us to include index i+2 and i+3 in the subsequence which was not possible in the previous approach.
2. Lastly don’t take the index and recursively call ‘i+1’, this case allows us to take indexes

‘I+1’ and ‘i+2’ in the subsequence which were not possible in above 2 cases.

**Code using 1-D DP :** [**https://ideone.com/d8BP93**](https://ideone.com/d8BP93)