DP Part 2

Soln is not important, knowing How to reach the soln is important

1) dp[] → N
2) find dp[1], dp[2], dp[3] by basic observations
3) Create a general formula
4) Now we can calculate dp[1]...dp[2]... using loop and formula.
5) In most cases dp[n] will be our final answer

Given an array a = [2, -3, 5, -8, 7], size = 5

Find Maximum Sum Subset such that no 2 elements are adjacent

[1, 2, 3, 4, 5, 6]

(2,4) (2,5) (3,4) (3,4,5) (3,4,6) (2,3) (2,4) (2,5)

(2,5,6) (2,5,7)

Choose a subset whose sum is max + its valid subset

dp[] = 5

dp[1] = best answer to question if size of array was 1

= 2 i.e a[1]

dp[2] = if size was 2 array would be like [2, -3]

= 2 as -3 is small we need max + no adjacent

dp[2] = 2

dp[3] = {2, -3, 5} adjacent ✗

ans can be {2, 5} or {-3, 5} or [2] or {-3} or {5}

we need max answer so {2, 5} = 7

dp[4] = {2, -3, 5, -8, 7} for size = 4 {2, -3, 5, -8}

we know dp[1] | dp[2] | dp[3]

let say we ~~know~~ include -8 in dp[4]

dp[4] = -8 + (we can't choose 5) so {2, -3}

= -8 + {best ans from (2, -3)}

= -8 + dp[2]

= a[4] + dp[2]

if we ignore -8 we choose from [2, -3, 5] then

dp[4] = dp[3]

so dp[4] = max (dp[3], -8 + dp[2])

dp[4] = max (dp[3], a[4] + dp[2])

for size = i

dp[i] = max (dp[i-1], a[i] + dp[i-2])

dp[5] = {2, -3, 5, -8, 7}

include 7 then dp[5] = 7 + dp[3]

= a[5] + dp[3]

not include 7

dp[5] = dp[4]

final ans dp[5] = max (dp[4], a[5] + dp[3])