Link - https://www.designa.in/17735/uber-oa-ctc-60-l-sde1-sep-5

-> Given an array of size "N" ; you have to do 3 type of operations sequentially

-> First delete any p elements;

-> Then delete any q-adjacent pair elements.

-> Then delete any r-adjacent triplet elements.

After doing all this :- Sum of the remaining array should be maximum.

Observation :-> If Q = 0 R = 0 and P!=0 -> solution is to remove p smallest elements in the array the remaining array would give us the maximum sum.

Observation :-> Order of operations does not matter -> you can do any operation anytime. Final result will be the same.

Easier version :- Q = 0 R = 0

How to do easier version with dp ->

dp[i][j] = best way to reach till index i by ignoring j elements;

=> dp[i][j] = max(dp[i-1][j-1] , b[i] + dp[i-1][j])

=> https://www.jdoodle.com/ia/1ghl

Medium - version :- R = 0

-> dp[i][j][k] = consider first "i" elements but ignore j elements ; and k pair elements.

-> dp[i][j][k] = max(dp[i-1][j-1][k] , b[i] + dp[i-1][j][k],dp[i-2][j][k-1])

Actual version :- P!=0 Q!=0 R!=0

-> dp[i][j][k][l] = consider first "i" elements but ignore j elements ; and k pair elements and l triplet elements;[all adjacent type;]

-> dp[i][j][k][l] = max(dp[i-1][j-1][k][l], b[i] + dp[i-1][j][k][l],dp[i-2][j][k-1][l],dp[i-3][j][k][l-1])

TC - O(NPQR)

Takes O(NPQR) size;

C++ https://www.jdoodle.com/ia/1gh7.

Java. https://www.jdoodle.com/ia/1gh8

Py https://www.jdoodle.com/ia/1ghb