Problem Statement Suggest Edit

You are given an array *'arr'* of *'n'* integers.

You have to divide the array into some subarrays such that each element is present in exactly one of the subarrays.

The length of each subarray should be at most *'k'*. After partitioning all the elements of each subarray will be changed to the maximum element present in that subarray.

Find the maximum possible sum of all the elements after partitioning.

arr : {1 15 7 9 2 5 10}

k = 3

Understanding question:

arr = {1, 15, 7, 9, 2, 5, 10}   K = 3

We have to part the array in sub-array such that every sub-array has size less than K and the every sub-array maxi m element will be replace by in other element of that sub-array.

for eg:

1 15 7 | 9 2 5 | 10

15 15 15 9 9 9 10

- We have to return the maximum sum of the all the sub array after replacement

eg:

$15 \times 3 + 9 \times 5 + 10 =$

4

$k < 5$

1, 15, 7, 9

$k < 5$

We use front partition algorithm.

left

1 | 15 7 9 2 5 10

1

f(i) → f(0) → Give me the maximum sum of sub-array starting from index 0

if (i == n) return 0;

len = 0, maxi = INT_MIN, → to handle if i+k exceed n

for(j = i; j < min(n, i+k) ; j++) len

len++;

maxi = max (maxi, arr[j]);

sum = (len x maxi) + f(j+1) ; → Give the maximum sum from index j+1.

maxSum = max (sum, maxSum); → maxSum

return maxSum;

Why len x maxi ?

len

1 15 7

maxi

15 15 15

maxi x len

15 × 3 = 45

for(j = 0 → k

T.C. : Recursion → Exponential (Overlaping Subproblem)

Memoization → dp[n] O(N × K)