

LEETCODE - (53) Maximum Sub Array.

→ Int. array nums, find subarray with largest sum, then return its sum.

Code: Java

class Solution {

public int maxSubArray (int [] nums) {

int cMax = nums[0]; → current max (jo iss index tak best hai)

int maxSum = nums[0]; → overall best till now.

for (int i=1; i < nums.length; i++) {

cMax = Math.max(nums[i], cMax + nums[i]);

maxSum = Math.max(cMax, maxSum);

}

return maxSum;

}

}

eg1- nums[] = [-2, 1, -3, 4, -1, 2, 1, -5, 4]

cMax
nums[i]

maxSum

at 0 -2 -2

i=1 → nums[1] = 1

cMax = (1, -2+1) max

= (1, -1) max. = (1)

∴ maxSum = max(1, -2) = (1)

i=2 → nums[2] = -3

cMax = max(-3, -2) = (-2)

maxSum = max(-2, 1) = (1)

i=3 → nums[3] = 4

cMax = max(4, 2) = (4)

maxSum = max(4, 1) = (4)

i=4 → nums[4] = -1

cMax = max(-1, 3) = (3)

maxSum = max(3, 4) = (4)

i=5 → nums[5] = 2

cMax = max(2, 5) = (5)

maxSum = max(5, 4) = (5)

$$i=6 \rightarrow \text{nums}[6] = 1$$

$$\text{maxSum} = \max(6, 5) = \textcircled{6}$$

$$c\text{max} = \max(1, 6) = \textcircled{6}$$

$$i=7 \rightarrow \text{nums}[7] = -5$$

$$\text{maxSum} = \max(1, 6) = \textcircled{6}$$

$$c\text{max} = \max(-5, 1) = \textcircled{1}$$

$$i=8 \rightarrow \text{nums}[8] = 4$$

$$\text{maxSum} = \max(5, 6) = \textcircled{6}$$

$$c\text{max} = \max(4, 5) = \textcircled{5}$$

$$\Rightarrow \text{maxSum} = 6$$

$$\Rightarrow \text{BestArray} = [4, -1, 2, 1]$$

$\Rightarrow \therefore$ This is called Kadane's Algorithm.