

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) {

Decide, naya subarray start karna hai, ya fir start wale ko extend karna how int cMax = nums[0]; → current max (jo iss index tak pware)

ab tak ka global best update karo. } return maxSum; → return kardouss subarray Ko, jo sum mei sabse kada hai }

}

eg) nums[] = [-2, 1, -3, 4, -1, 2, 1, -5, 4].

cmax	nums[i]	maxSum

at 0 -2 -2 ∵ maxSum = max(1, -2) = (1).

i=1 → nums[1] = 1. cMax = (1, -2+1) max
= (1, -1) max. = (1)

i=2 → nums[2] = -3. maxSum = max(-2, 1) = (1).

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

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

i=3 → nums[3] = 4 cMax = max(4, 2) = (4)

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

i=4 → nums[4] = -1 cMax = max(-1, 3) = (3)

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

i=5 → nums[5] = 2 cMax = max(2, 5) = (5)

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

$$i=6 \rightarrow \text{num}[6] = 1 \quad \text{maxSum} = \max(6, 5) = 6$$

$$cMax = \max(1, 6) = 6$$

$$i=7 \rightarrow \text{num}[7] = -5 \quad \text{maxSum} = \max(1, 6) = 6$$

$$cMax = \max(-5, 1) = 1$$

$$i=8 \rightarrow \text{num}[8] = 4 \quad \text{maxSum} = \max(5, 6) = 6$$

$$cMax = \max(4, 5) = 5$$

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

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

\Rightarrow This is called Kadane's Algorithm.

$$\text{Ans} = [\text{LeftIndex}, \text{RightIndex}] = [1, 4]$$

$$(1) = 6 + (-1) + 2 + 1 = 8$$

$$(1) = 6 + (-1) + 2 + 1 = 8$$

$$(1) = 6 + (-1) + 2 + 1 = 8$$

$$(1) = 6 + (-1) + 2 + 1 = 8$$

$$(1) = 6 + (-1) + 2 + 1 = 8$$