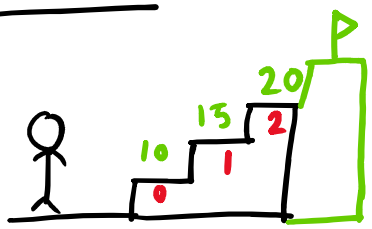
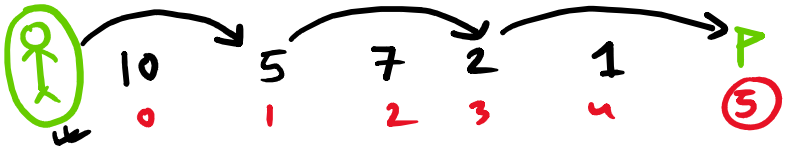
Dynamic Programming: Staircase and Kadane's Algorithm

Saturday, December 27, 2025 2:08 PM

Staircase



cost = 15



5 + 2 + 0 = 7

Defination

f(step) = min cost to reach last index from step + including cost of step

(0 - 5)

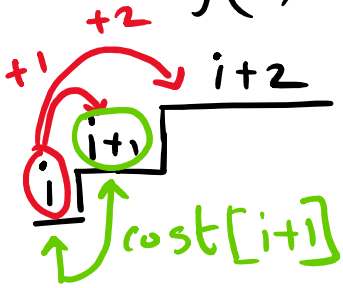
Transition

$$f(i) = cost[i] + \min \begin{cases} f(i+1) \\ f(i+2) \end{cases}$$

cost[i]

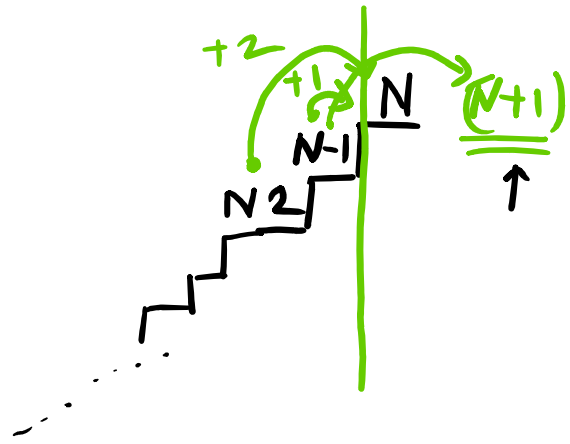
cost[i+1]

cost[i+2]



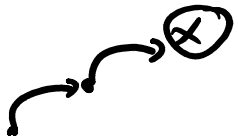
Base Case

$i \geq N$ $\quad f(i) = 0$ ←



① 1 — x → 1, 2, 3, 4, ... , x

② [1, 5, 10] →

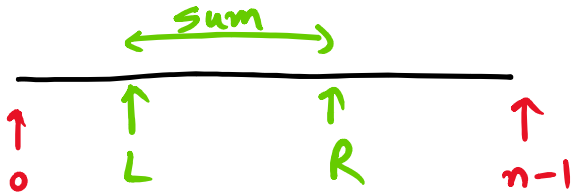


Kadane's Algorithm

O(N³)

1	-2	5	-3	7	-10
0	1	2	3	4	5

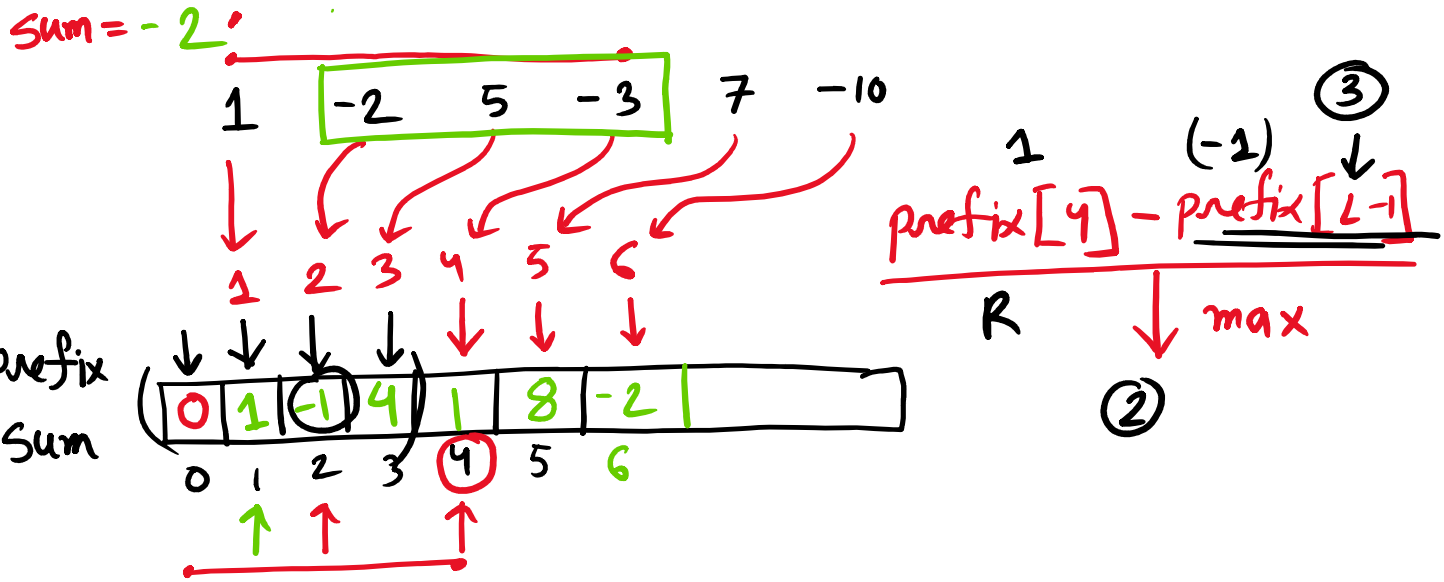
M.S.S = 9

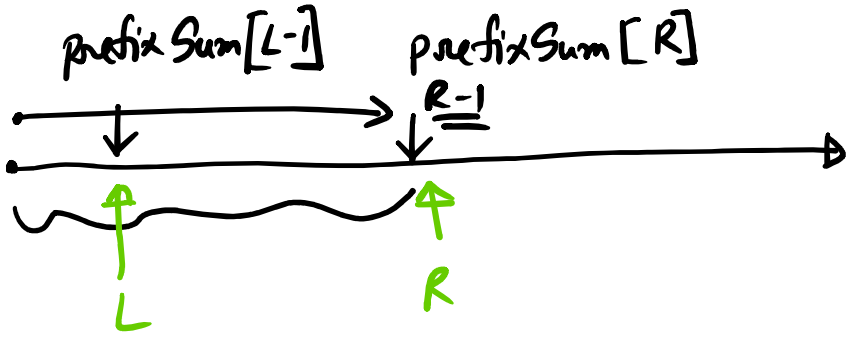


result =

```
for(L = 1 - N) {  ⟶  L = 1, 2, 3, ...., N-1
    for(R = L - N) {     R = L, L+1, L+2, ...., N-1
        for( i → L-R) {
            sum += nums[i]
        }
        result = max(result, sum)
```

```
            result = max(result, sum)
        }
    }
    return result
```





prefix[R] − prefix[L-1] = MAX

min