

Dynamic Programming

Maximum Subarray Sum

MSS works by calculating the MSS of the subarray that ends at index i (the current index). This subarray is then used to calculate the MSS of the next index $i+1$. At each iteration, $MSS[i] = \max(\text{original_array}[i], \text{original_array}[i] + MSS[i-1])$. The final answer is the max value of MSS at any index. Below you can see the progress of MSS through an array and the final selection.

0	1	2	3	4	
7	8	5	2	-1	original array
7	15	0	0	0	mss progress

0	1	2	3	4	
7	8	5	2	-1	original array
7	15	20	0	0	mss progress

0	1	2	3	4	
7	8	5	2	-1	original array
7	15	20	22	0	mss progress

0	1	2	3	4	
7	8	5	2	-1	original array
7	15	20	22	21	mss progress

0	1	2	3	4	
7	8	5	2	-1	original array
7	15	20	22	21	mss progress

Complexity

Time complexity: $O(n)$ where n is the number of elements in the array

- It takes $O(1)$ time to calculate each element of the MSS array which has n elements total

Space complexity: $O(n)$