# Week 12 : Problem solving Strategies

1. Maximum subarray problem.

Given an array $A[0 \ldots n-1]$, find a contiguous subarray $A[i..j]$ which has the largest sum.

E.g. $A = \{1, 3, 2, 7\}$

$\text{max\_subarray}(A) = \{1, 3, 2, 7\}$

$A = \{5, -3, 4, -1\}$

$\text{max\_subarray}(A) = \{5, -3, 4\}$

- Kaden's Algorithm

- Logic : If max subarray sum ending at index $i$ is $S_i$, what is the max subarray sum ending at position $i+1$ i.e., $S_{i+1}$

$$S_{i+1} = \max(S_i + A_{i+1}, A_{i+1})$$

- Observation : Each time when we add an element $A_{i+1}$ to $S_i$, if $S_{i+1}$ becomes $< 0$, $A_{i+1}$ can not be a part of the solution. [Unless all elements $< 0$]

Eg : $\{\underbrace{5, 2}, -8, \underbrace{3}\}$

- max. subarray sum ending at position i+1 ( $S_{i+1}$ ) either includes $S_i$ or it doesn't.

## Algorithm

```
max_subarray(A)
        1. sum = max_sum = A[0]
        2. for i = 1 to n-1
                sum = max (sum, sum+ A[i])
            max_sum = max (sum, max_sum)
        3. Return max_sum.
```