Maximum-sum contiguous subarray

Maximum-sum contiguous subarray

Given an array A[ ] of n numbers, find the maximum sum found in any contiguous subarray

A zero length subarray has maximum 0

Example:

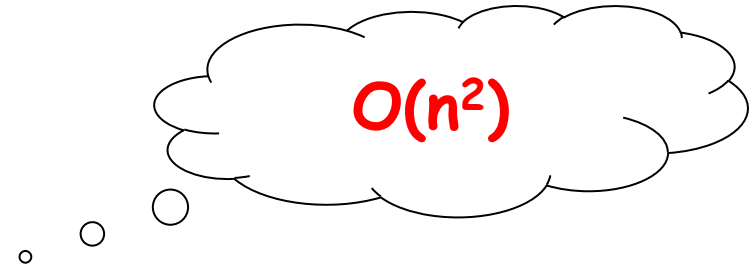
- if A[ ] contains **-2, 5, -3, 6, -1**, then the maximum sum will be 5-3+6 = **8**
- if A[ ] contains **10, 15, -3, -4, -2, -1, 8, 5**, then the maximum sum will be **28** (all numbers added together)

$O(n^2)$, $O(n \log n)$, $O(n)$ time algorithms

Brute-force algorithm

Brute-force algorithm

- All possible contiguous subarrays
  - A[1..1], A[1..2], A[1..3], ..., A[1..(n-1)], A[1..n]
  - A[2..2], A[2..3], ..., A[2..(n-1)], A[2..n]
  - ...
  - A[(n-1)..(n-1)], A[(n-1)..n]
  - A[n..n]



- How many of them in total?
- Algorithm: For each subarray, compute the sum. Find the subarray that has the maximum sum.

Idea of brute-force algorithm

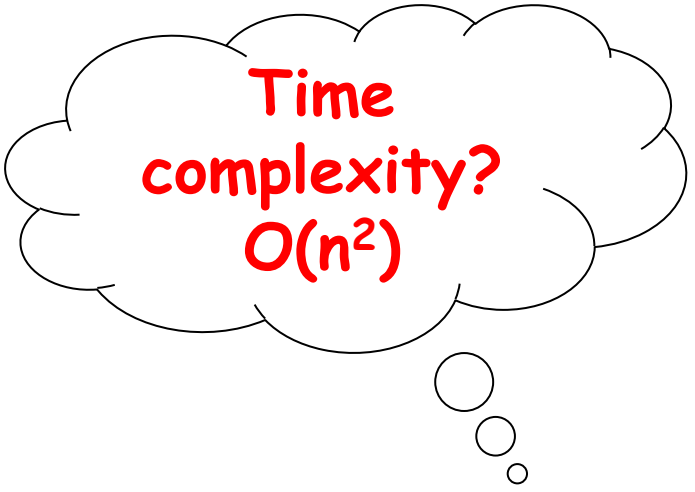
Example:	2	-6	-1	3	-1	2	-2
sum from A[1]:	2	-4	-5	-2	-3	-1	-3
sum from A[2]:		-6	-7	-4	-5	-3	-5
sum from A[3]:			-1	2	1	3	1
sum from A[4]:				3	2	4	2
sum from A[5]:					-1	1	-1
sum from A[6]:						2	0
sum from A[7]:							-2

Brute-force algorithm

- Outer loop: index variable i to indicate start of subarray, for $1 \le i \le n$, i.e., $A[1], A[2], \ldots, A[n]$
  - for i = 1 to n do ...
- Inner loop: for each start index i, we need to go through $A[i..i], A[i..(i+1)], \ldots, A[i..n]$
  - use an index j for $i \le j \le n$, i.e., consider $A[i..j]$
  - for j = i to n do ...

Pseudo code

```
max = 0
for i = 1 to n do
begin
    sum = 0
    for j = i to n do
    begin
        sum = sum + A[j]
        if sum > max
        then max = sum
    end
end
```



Divide-and-conquer algorithm

Divide-and-conquer algorithm

- maximum contiguous subarray in A[1..n], let **m=n/2**
  a. max contiguous subarray in A[1..m]
  b. max contiguous subarray in A[m+1..n]
  c. max contiguous subarray that spans across A[m]
- (a) & (b) can be found **recursively**
- (c) can be found in two steps
  - consider A[i..m] by fixing m but change i, i.e., $1 \le i \le m$
  - consider A[(m+1)..j] by fixing (m+1) but change j, i.e., $m+1 \le j \le n$
  - sum these two maximum

Idea of divide-and-conquer

Example: 10 15 -3 -4 | -2 -1 8 5

max on L (recursion) 10 15

max on R (recursion) 8 5

mid extend to L 10 15 -3 -4

mid extend to R -2 -1 8 5

- Possible candidates:
  - 25, 13, 28 (=18+10)
  - overall maximum **28**, i.e., take all numbers

Idea of divide-and-conquer

Example: -2 5 -1 | -5 2 -1 2

max on L (recursion) 5

max on R (recursion) 2 -1 2

mid extend to L 5 -1

mid extend to R *not take any*

- Possible candidates:
  - 5, 3, 4 (=4+0)
  - overall maximum **5**

Pseudo code

```
Algorithm MaxSum(p, q)
    if p == q then return max(A[p], 0)
    m = ⌊(p+q)/2⌋
    maxL = MaxSum(p,m)
    maxR = MaxSum(m+1, q)
    compute maxMidL
    compute maxMidR
    maxMid = maxMidL + maxMidR
    return max(maxL, maxR, maxMid)
```

Pseudo code

```
Algorithm MaxSum(p, q)
    if p == q then return max(A[p], 0)
    m = ⌊(p+q)/2⌋
    maxL = MaxSum(p,m)
    maxR = MaxSum(m+1, q)
    compute maxMidL
    compute maxMidR
    maxMid = maxMidL + maxMidR
    return max(maxL, maxR, maxMid)
```

```
sum = 0, maxMidL = 0
for i = m down to p
begin
  sum += A[i]
  maxMidL = max(maxMidL, sum)
end
```

Pseudo code

```
Algorithm MaxSum(p, q)
    if p == q then return max(A[p], 0)
    m = ⌊(p+q)/2⌋
    maxL = MaxSum(p,m)
    maxR = MaxSum(m+1, q)
    compute maxMidL
    compute maxMidR
    maxMid = maxMidL + maxMidR
    return max(maxL, maxR, maxMid)
```

```
sum = 0, maxMidR = 0
for i = m+1 to q
begin
  sum += A[i]
  maxMidR = max(maxMidR, sum)
end
```

Maximum Subarray Problem Example

Given an integer array nums, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

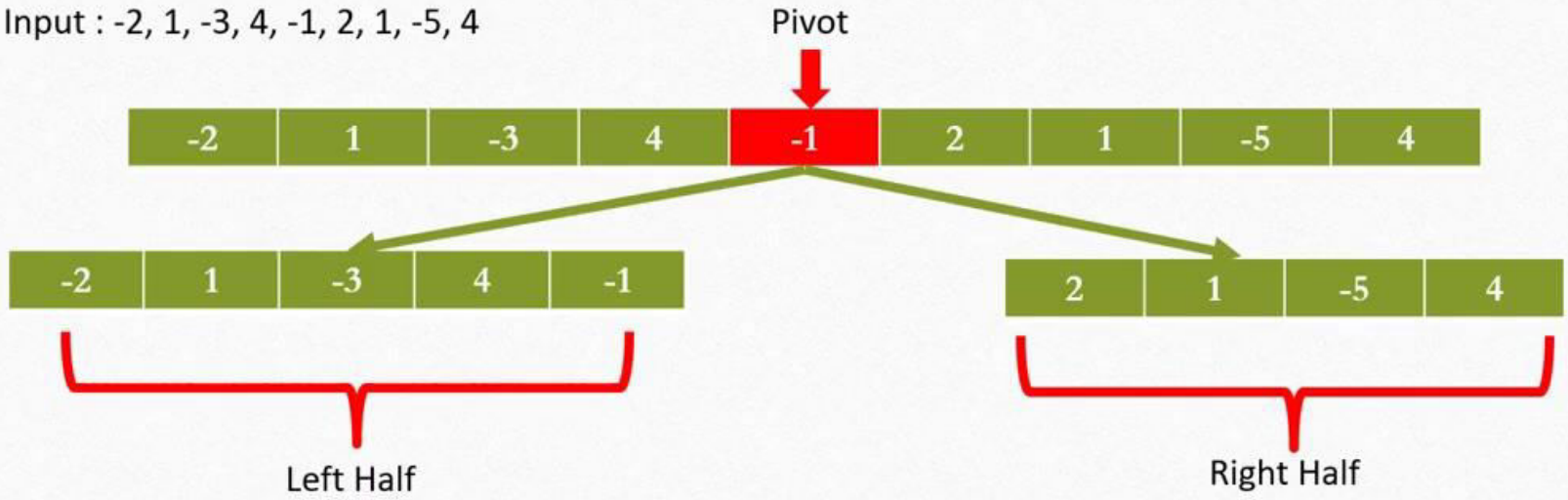
Example:

Input: [-2,1,-3,4,-1,2,1,-5,4],

Output: 6

Explanation: [4,-1,2,1] has the largest sum = 6.

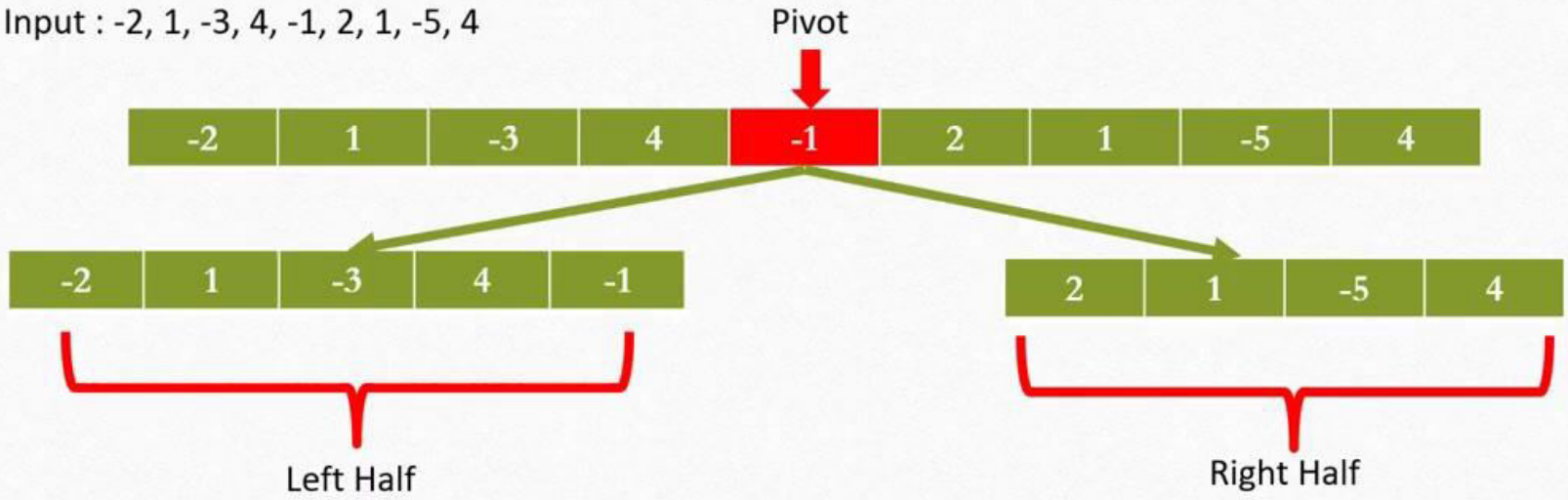
Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



$$\text{Index of Pivot} = \frac{(\text{begin} + \text{end})}{2} = \frac{(0+8)}{2} = 4$$

Left Half = begin to pivot (inclusive)

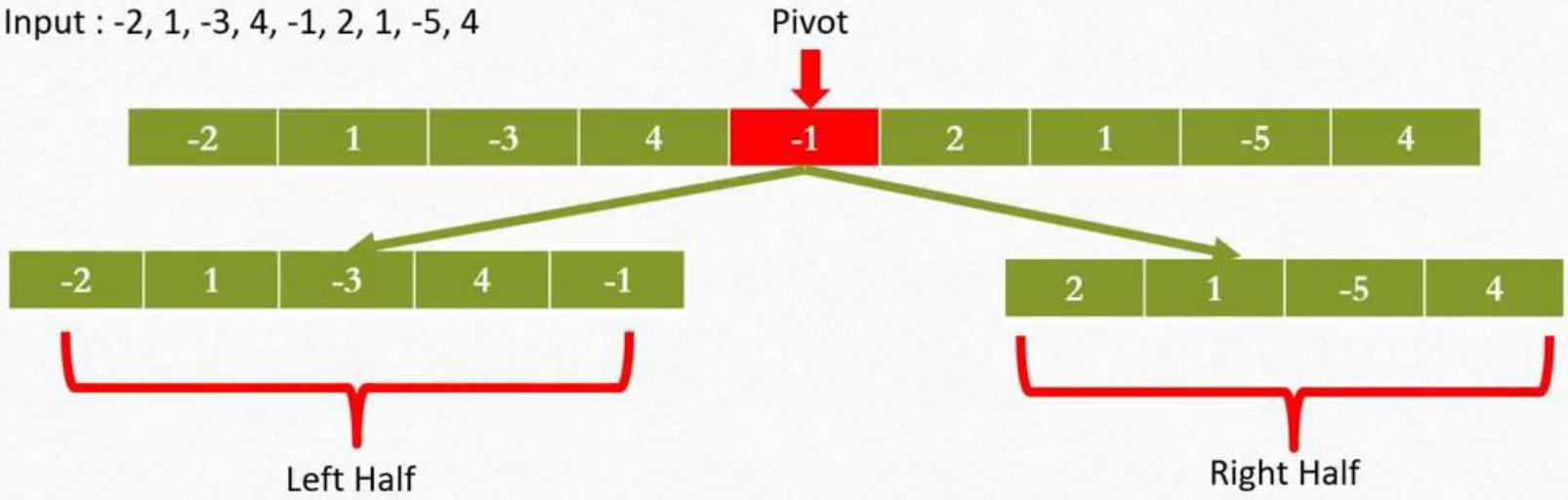
Right Half = pivot +1 to end



$$\text{Index of Pivot} = \frac{(\text{begin} + \text{end})}{2} = \frac{(0+8)}{2} = 4$$

Left Half = begin to pivot (inclusive)

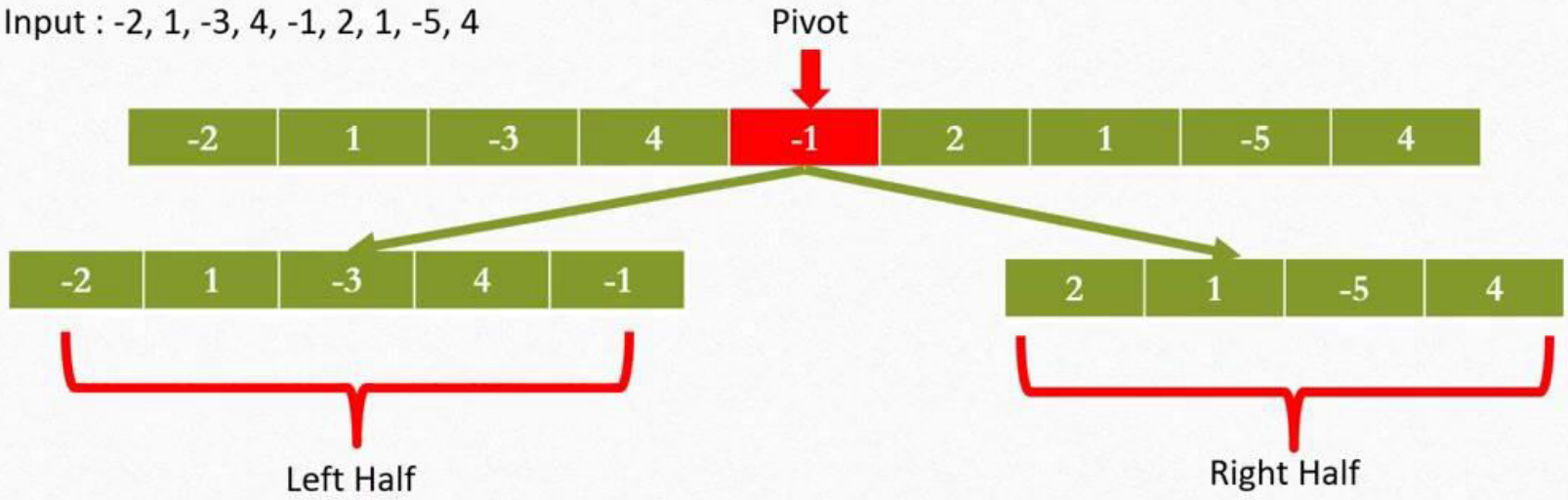
Right Half = pivot +1 to end



$$\text{Index of Pivot} = \frac{(\text{begin} + \text{end})}{2} = \frac{(0+8)}{2} = 4$$

Left Half = begin to pivot (inclusive)

Right Half = pivot +1 to end

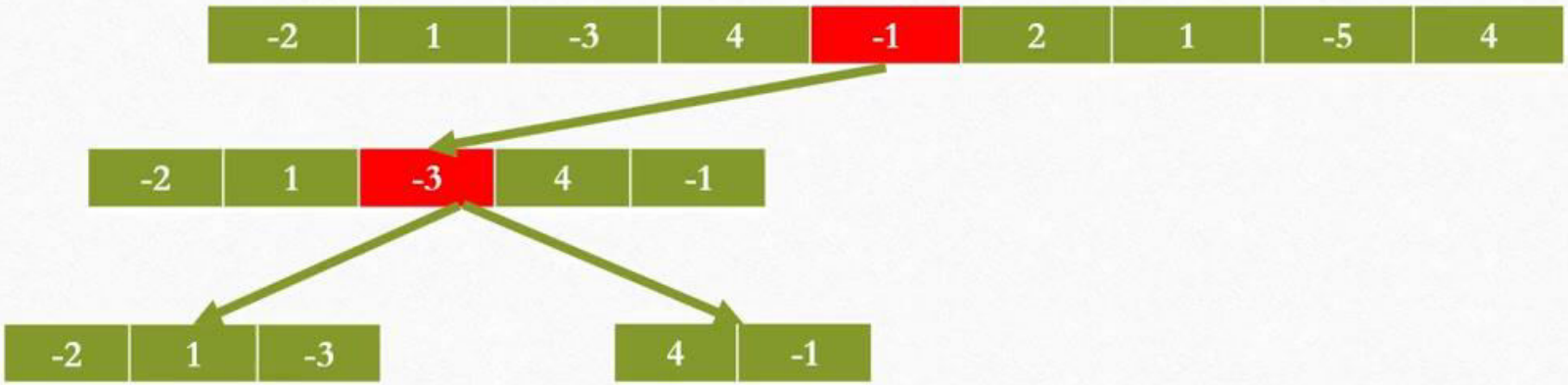


$$\text{Index of Pivot} = \frac{(\text{begin} + \text{end})}{2} = \frac{(0+8)}{2} = 4$$

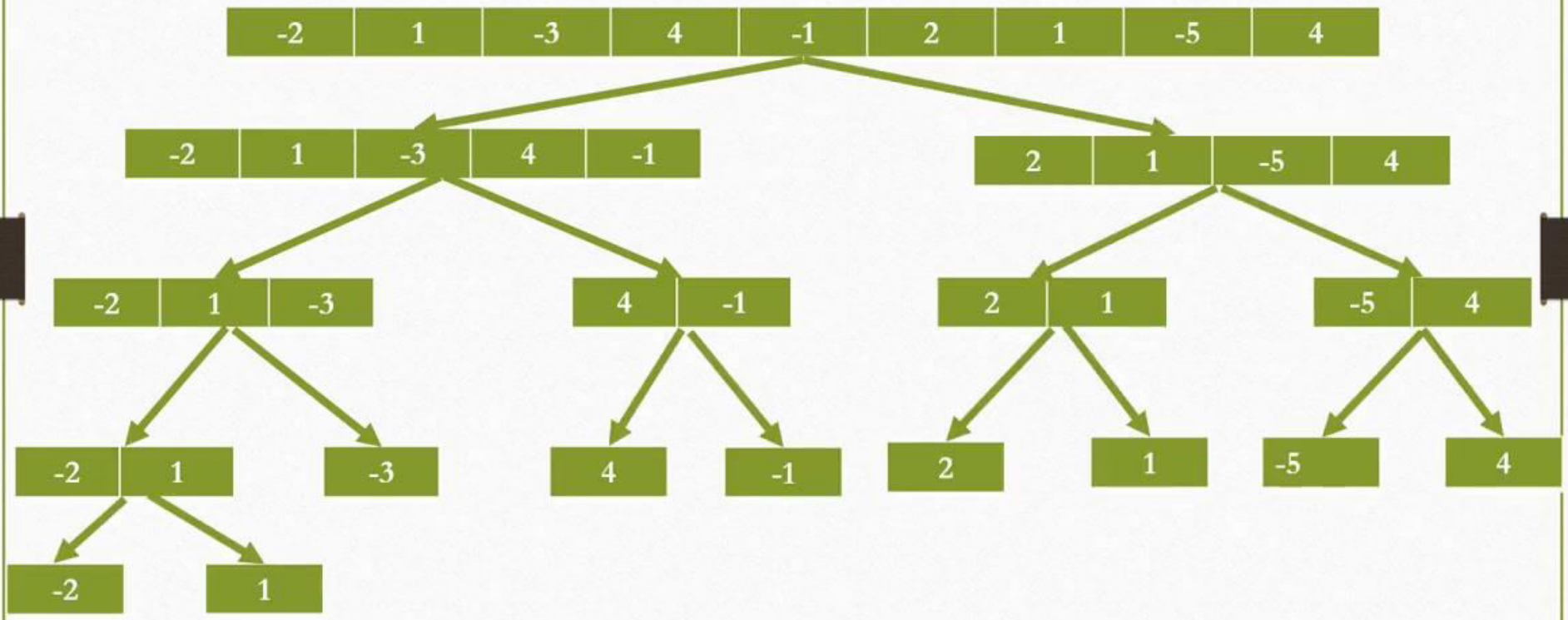
Left Half = begin to pivot (inclusive)

Right Half = pivot +1 to end

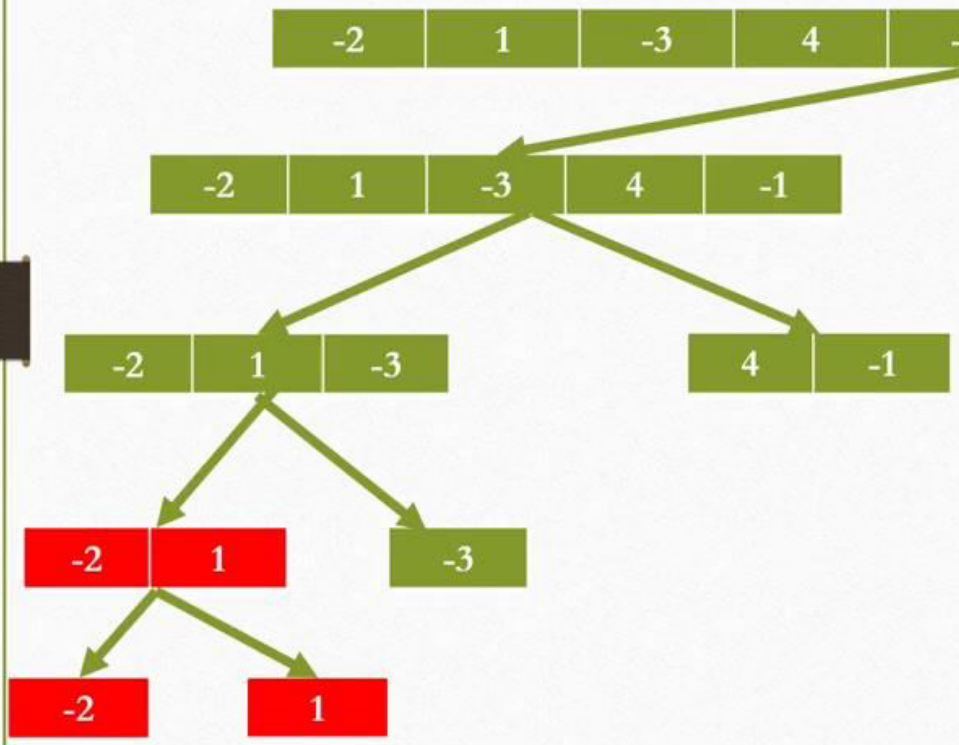
Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

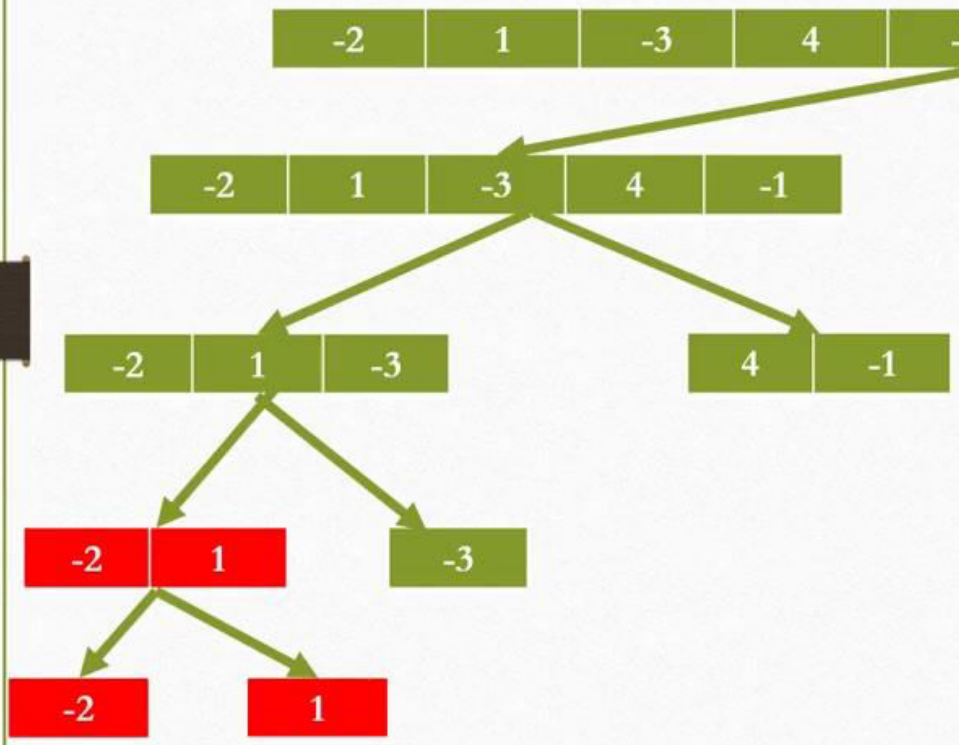
Right Sub Sum = 1

Cross Sum = Left Sub Sum +Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

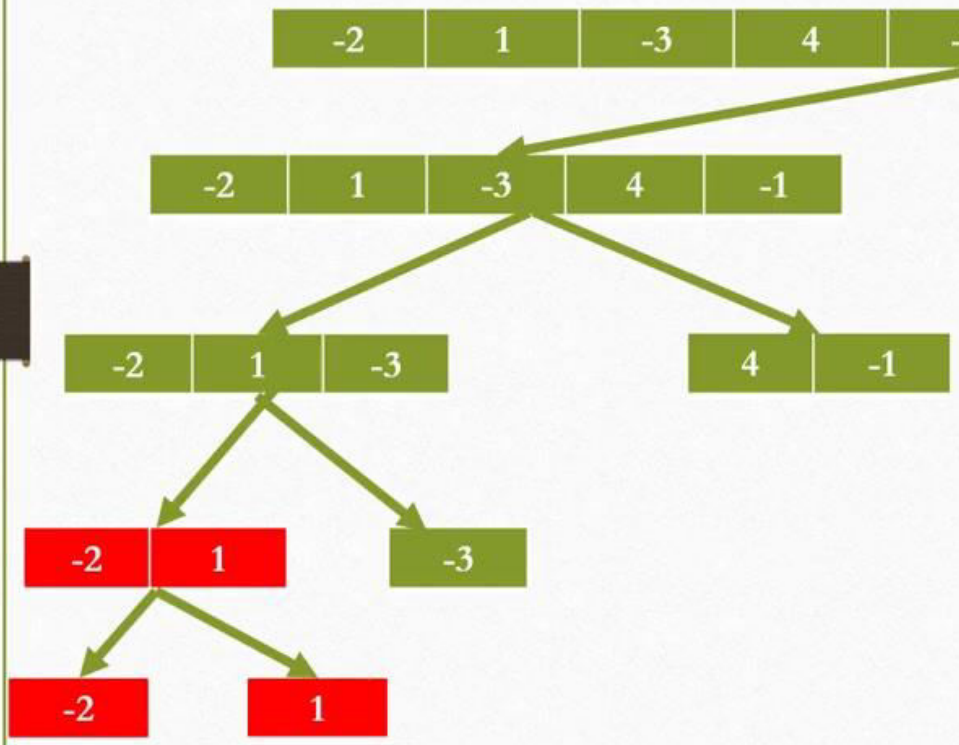
Right Sub Sum = 1

Cross Sum = Left Sub Sum +Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

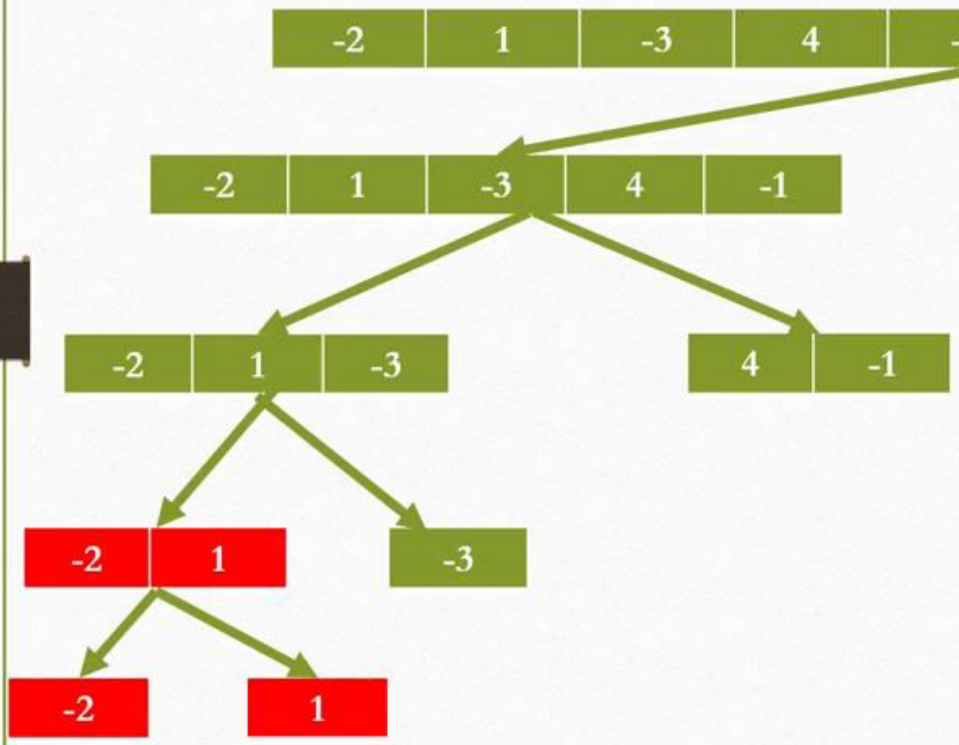
Right Sub Sum = 1

Cross Sum = Left Sub Sum +Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

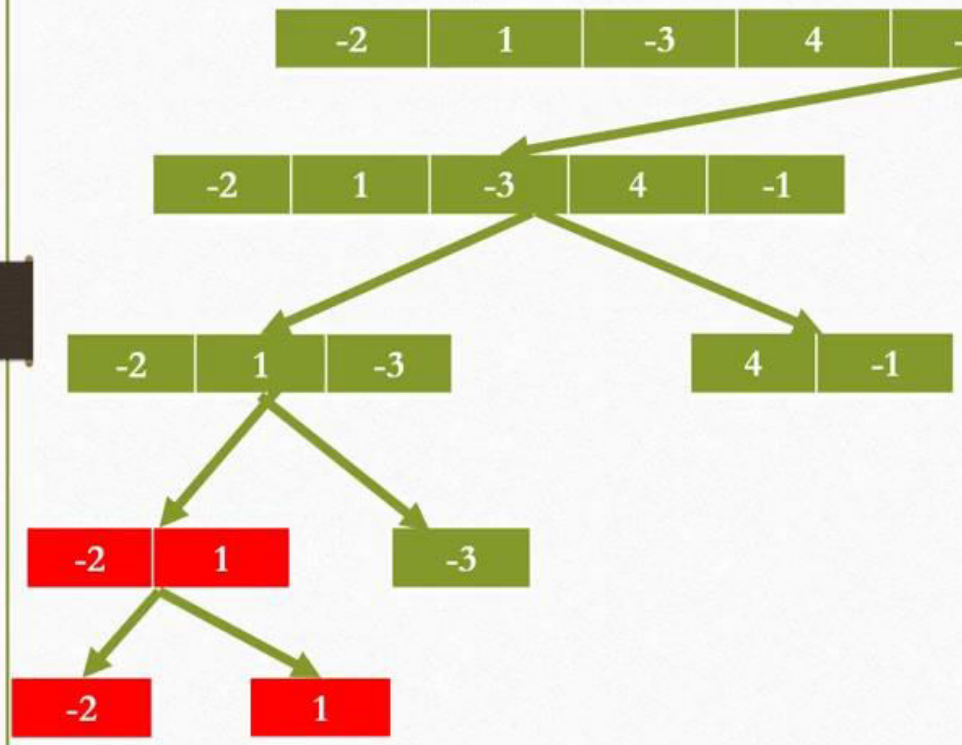
Right Sub Sum = 1

Cross Sum = Left Sub Sum +Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

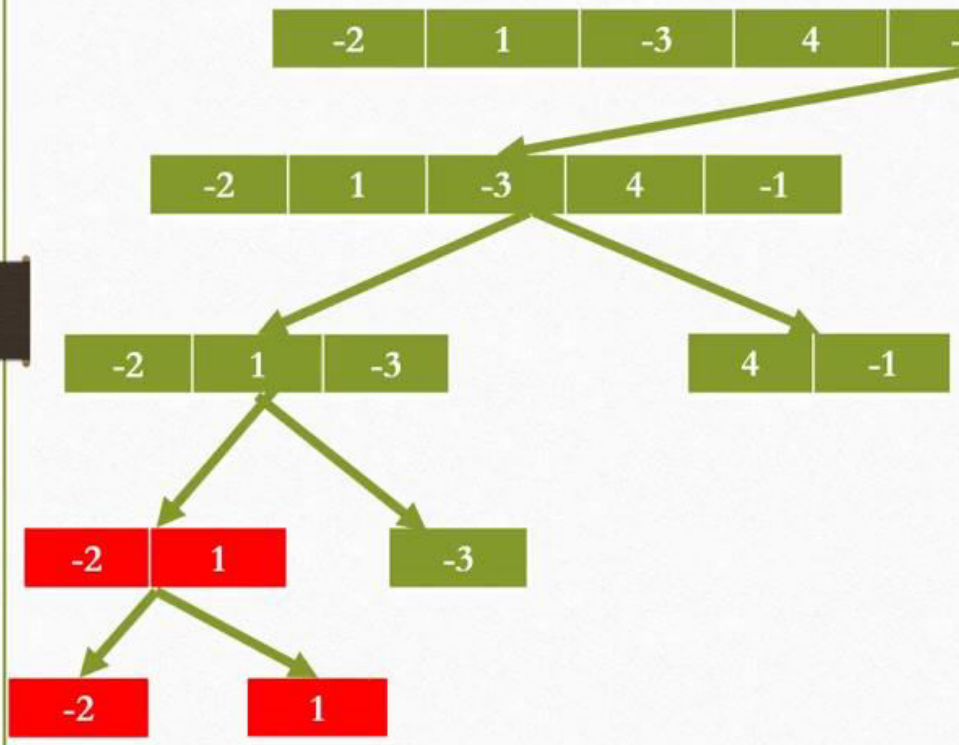
Right Sub Sum = 1

Cross Sum = Left Sub Sum +Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

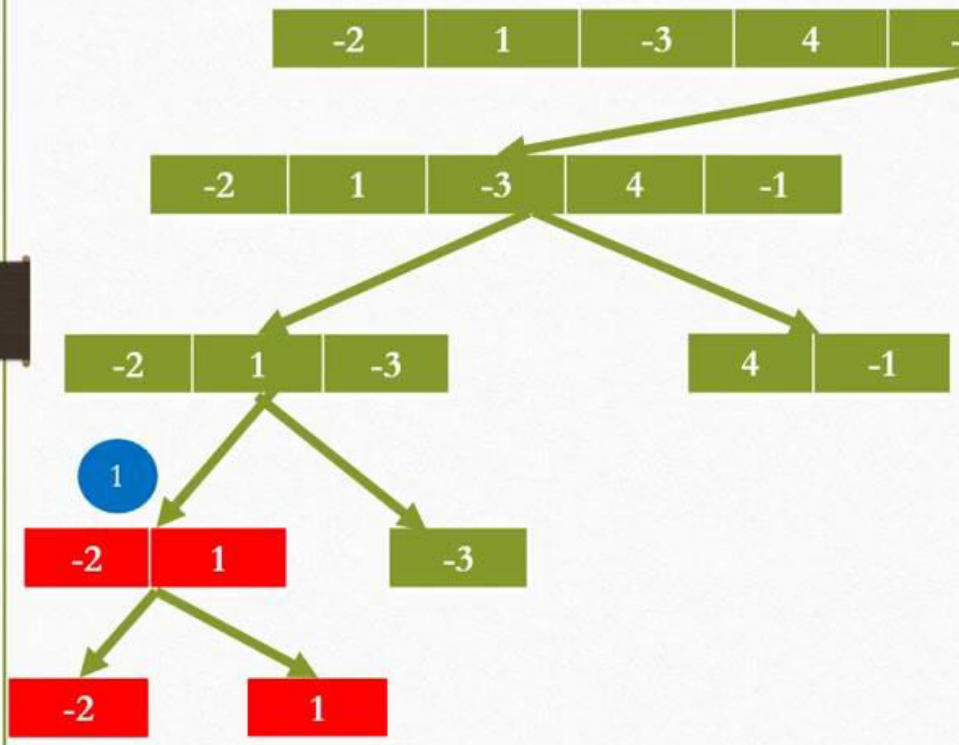
Right Sub Sum = 1

Cross Sum = Left Sub Sum +Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

Right Sub Sum = 1

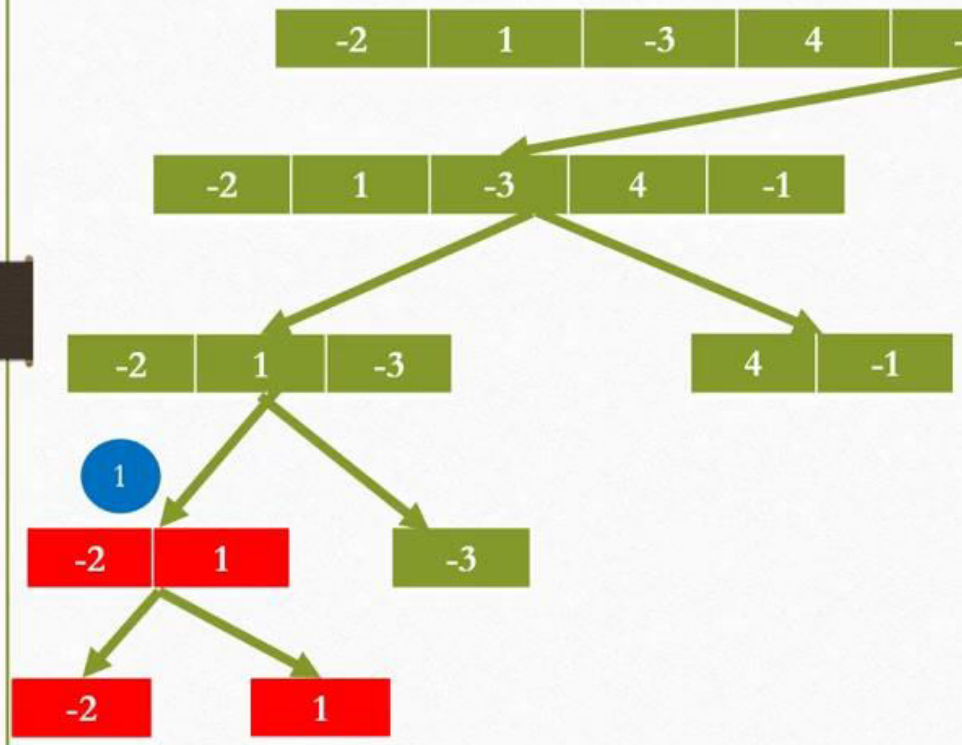
Cross Sum = Left Sub Sum + Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

Right Sub Sum = 1

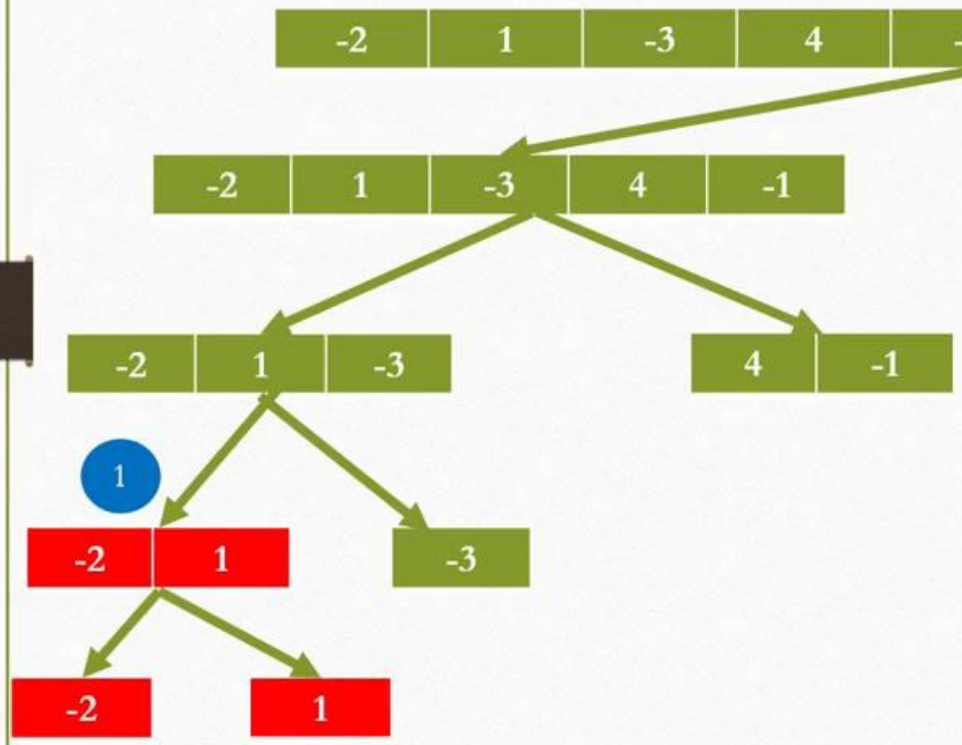
Cross Sum = Left Sub Sum + Right Sub Sum

= (-2+1) = -1

Cross Sum = -1

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -2

Left Sum = -2

Right Sum = 1

Cross Sum Calculations :

Left Sub Sum = -2

Right Sub Sum = 1

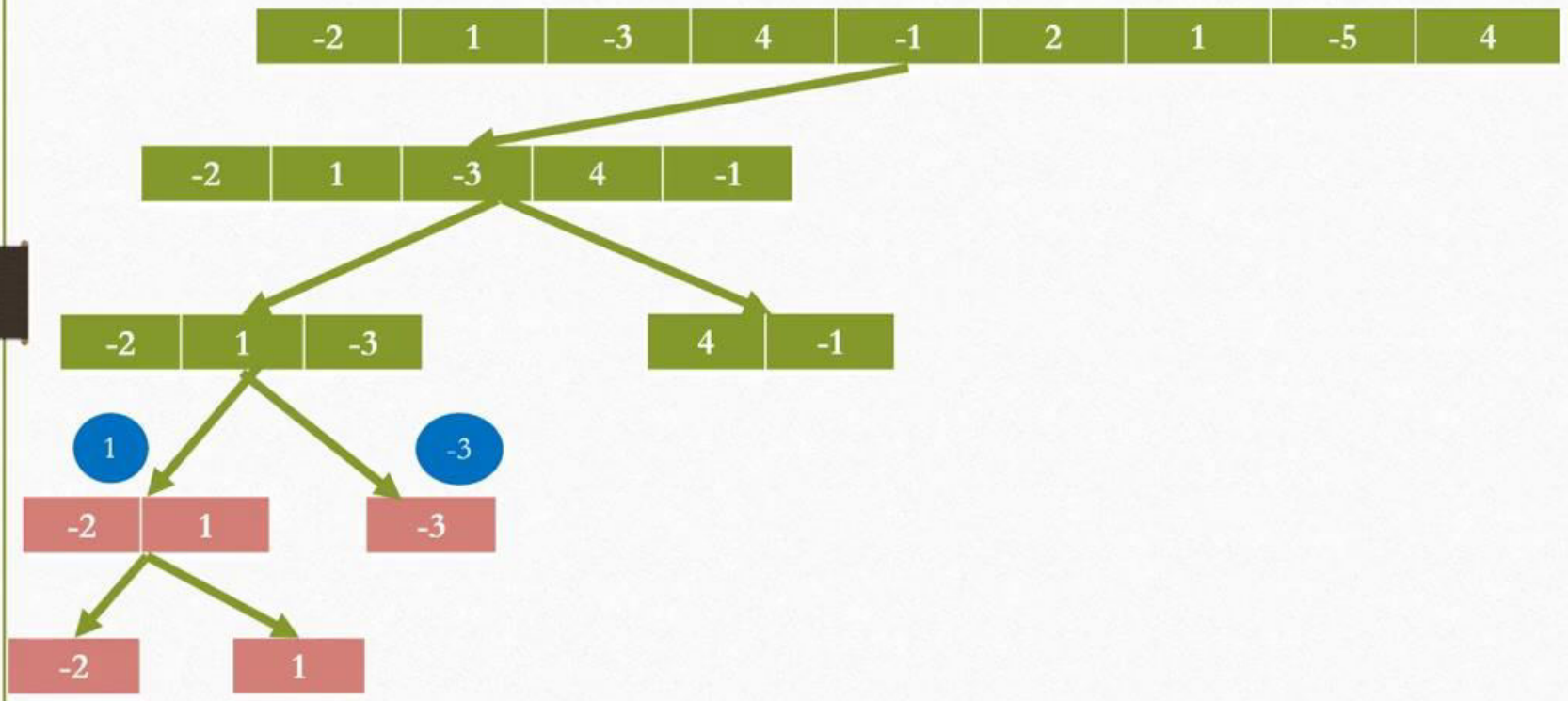
Cross Sum = Left Sub Sum + Right Sub Sum

= (-2+1) = -1

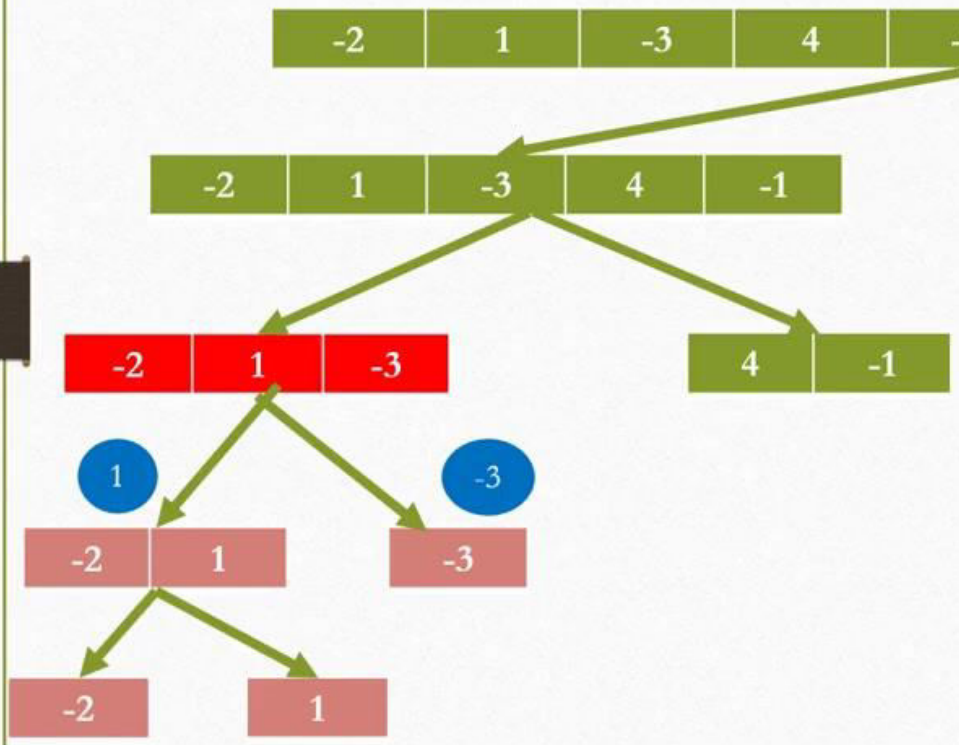
Cross Sum = -1

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(1, 1-2) = 1

Right Sub Sum = -3

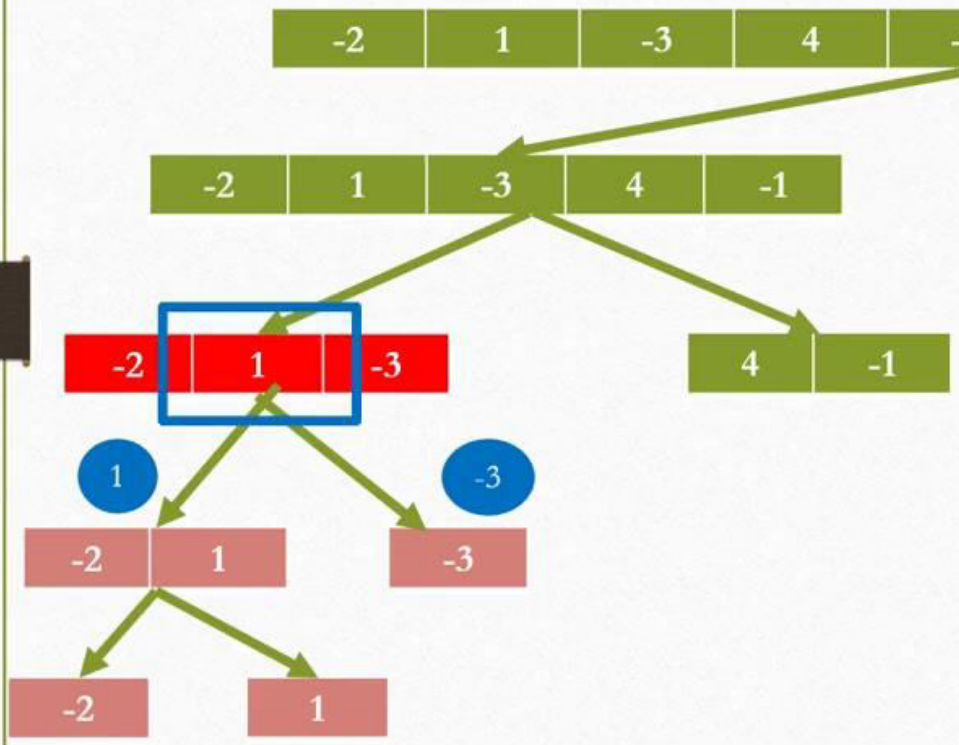
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(1, 1-2) = 1

Right Sub Sum = -3

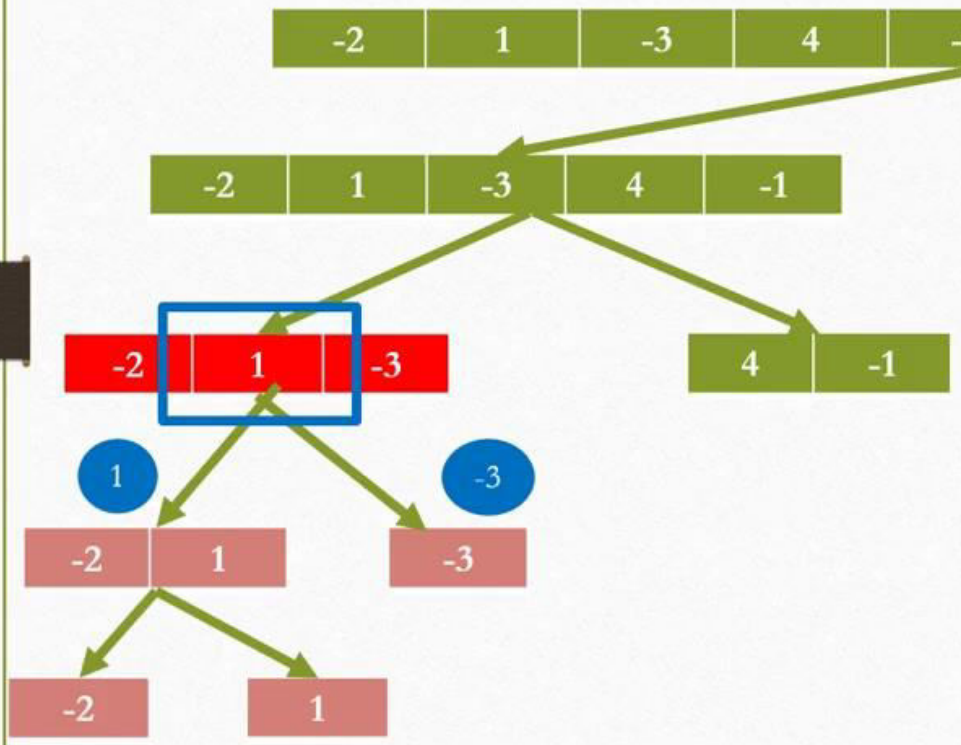
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(**1**, 1-2) = 1

Right Sub Sum = -3

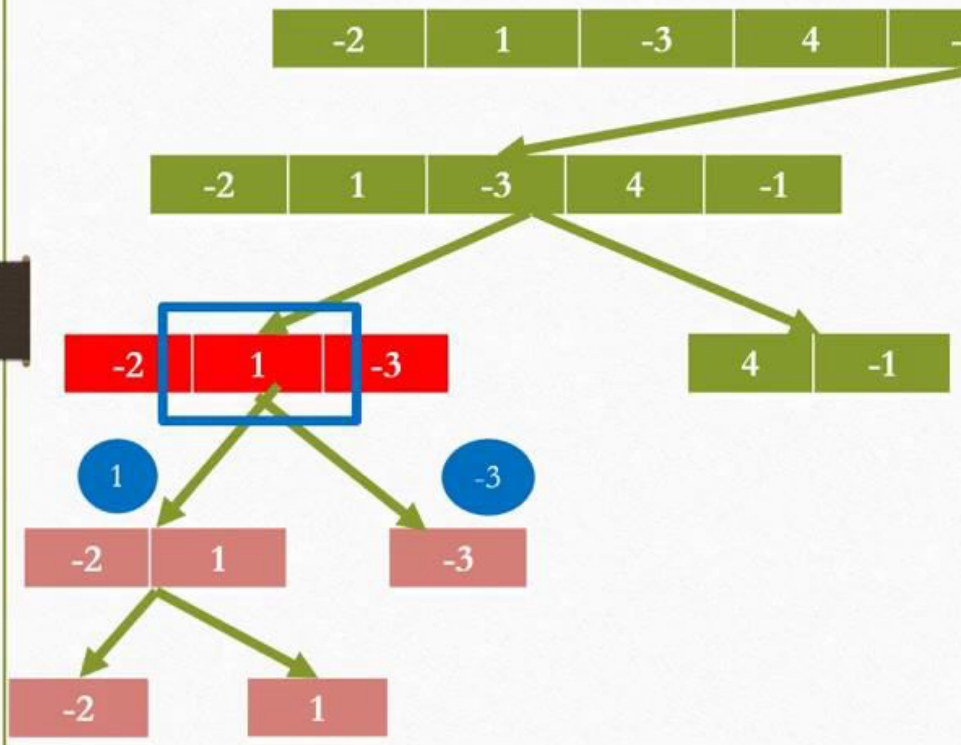
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(1, 1-2) = 1

Right Sub Sum = -3

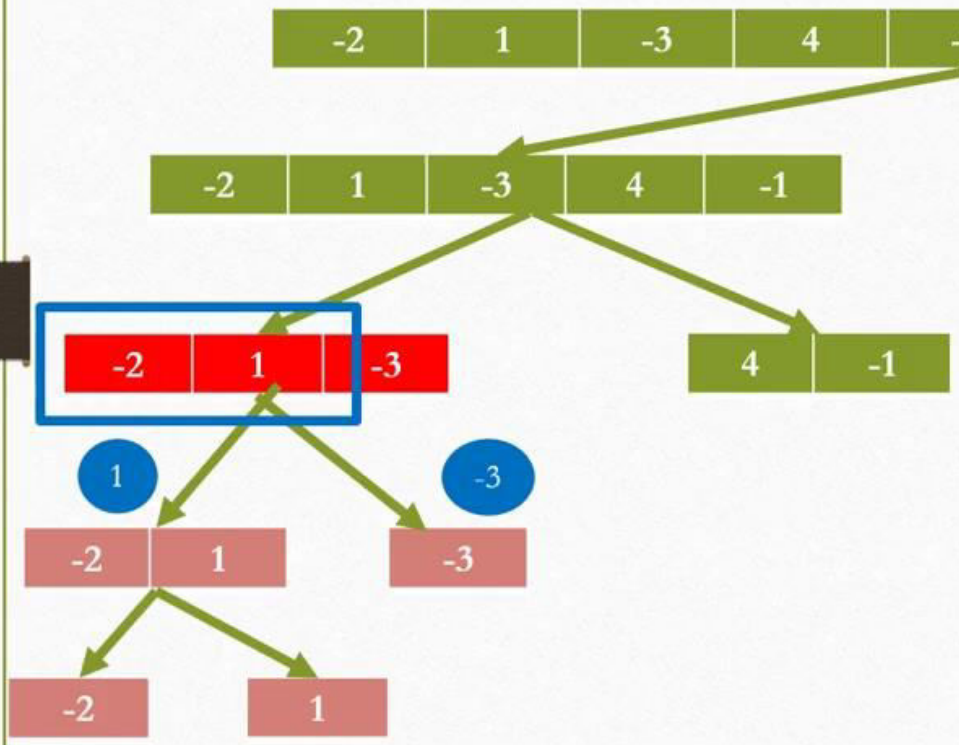
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(1, **1-2**) = 1

Right Sub Sum = -3

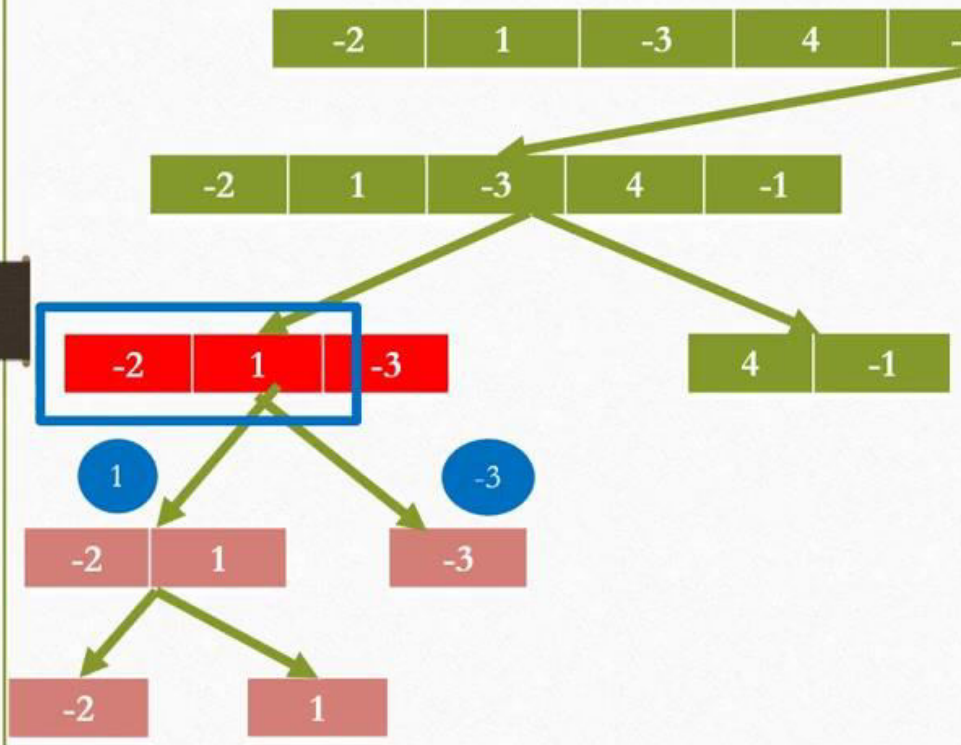
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(1, **1-2**) = 1

Right Sub Sum = -3

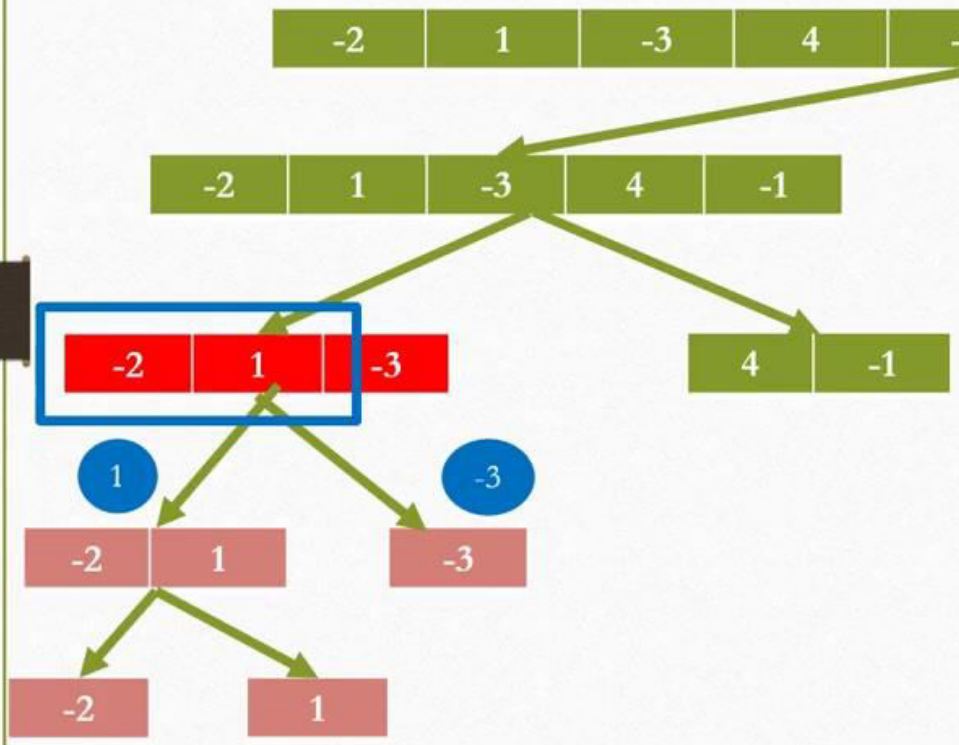
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(1, **1-2**) = 1

Right Sub Sum = -3

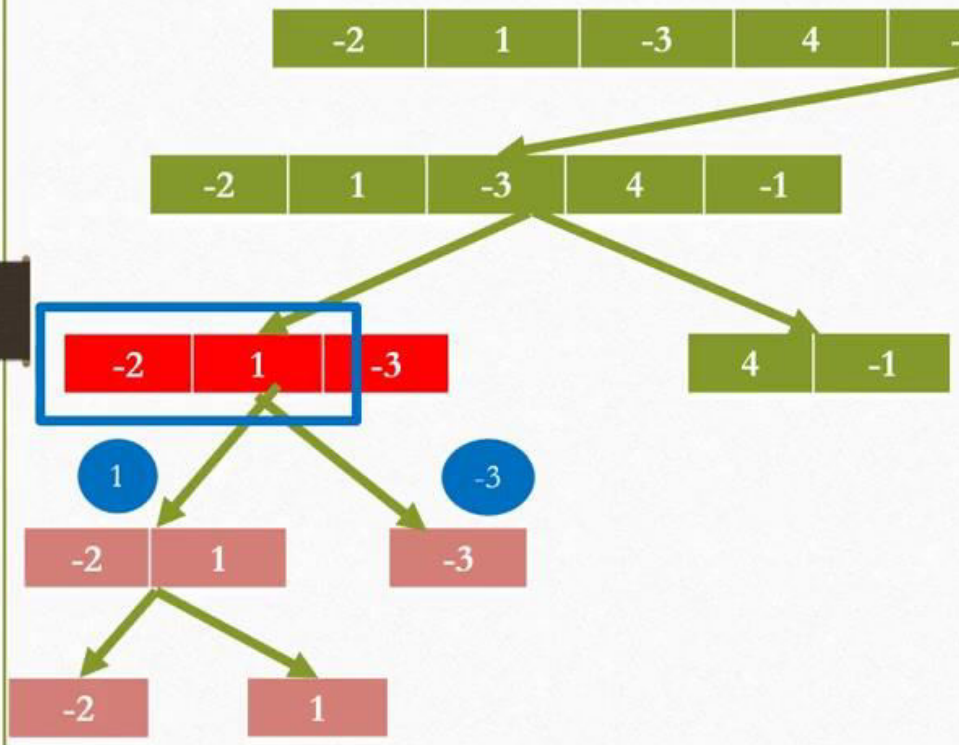
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 1

Left Sum = 1

Right Sum = -3

Cross Sum Calculations :

Left Sub Sum = Max(1, **1-2**) = 1

Right Sub Sum = -3

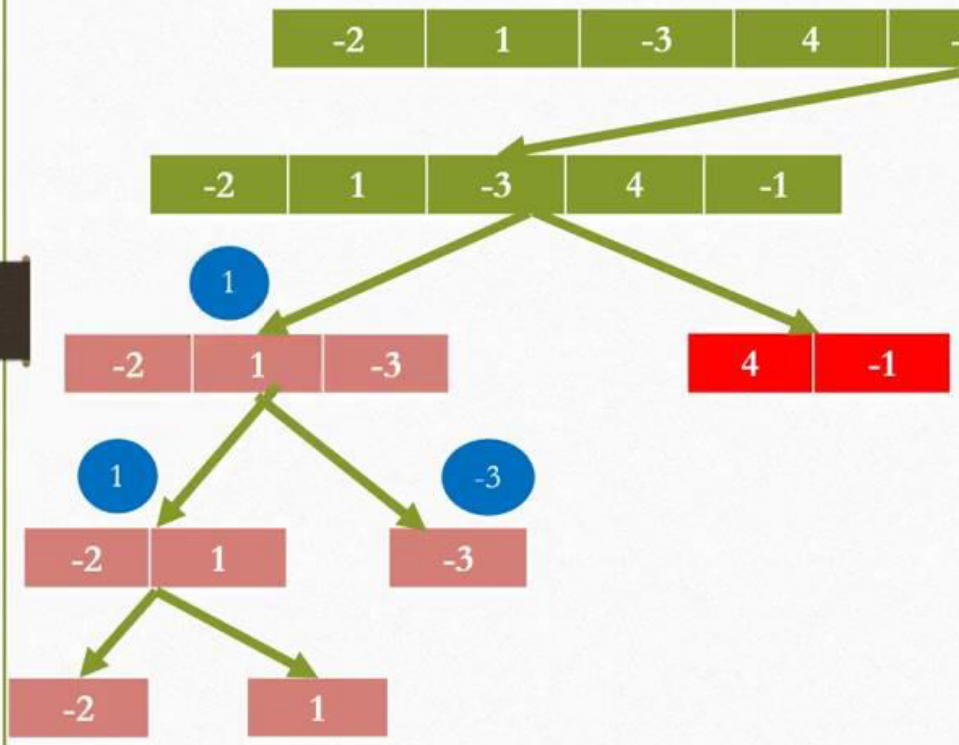
Cross Sum = Left Sub Sum + Right Sub Sum

= (1+ -3) = -2

Cross Sum = -2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 1

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = 4

Left Sum = 4

Right Sum = -1

Cross Sum Calculations :

Left Sub Sum = 4

Right Sub Sum = -1

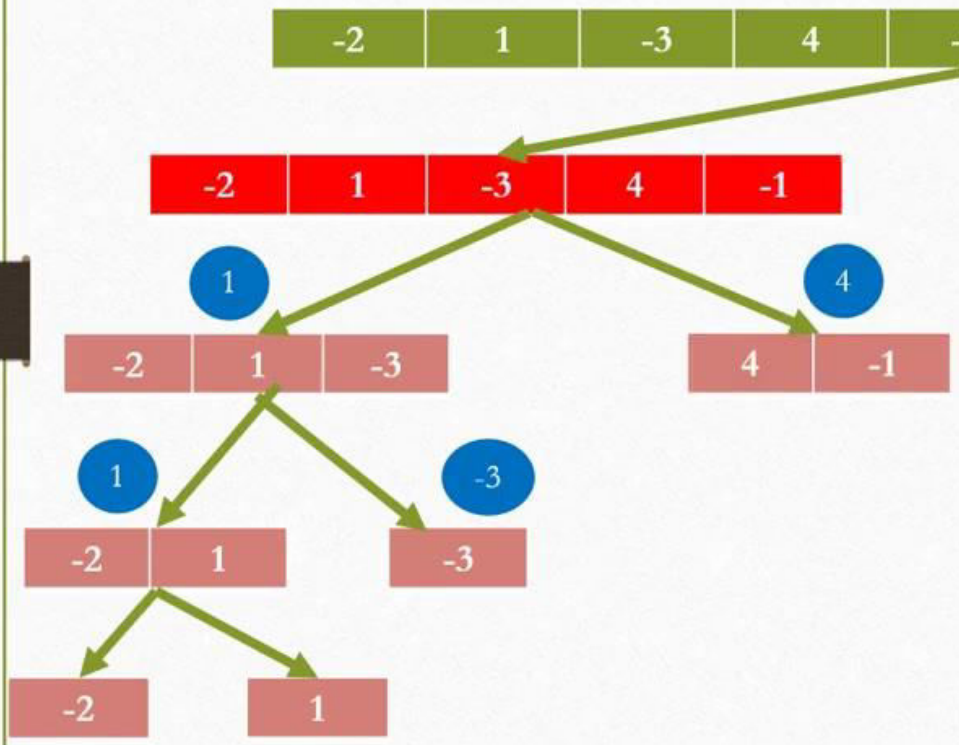
Cross Sum = Left Sub Sum+Right Sub Sum

= (4+ -1) = 3

Cross Sum = 3

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 4

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -3

Left Sum = 1

Right Sum = 4

Cross Sum Calculations :

Left Sub Sum = **Max(-3, -3+1, -3+1-2) = -2**

Right Sub Sum = **Max(4, 4-1) = 4**

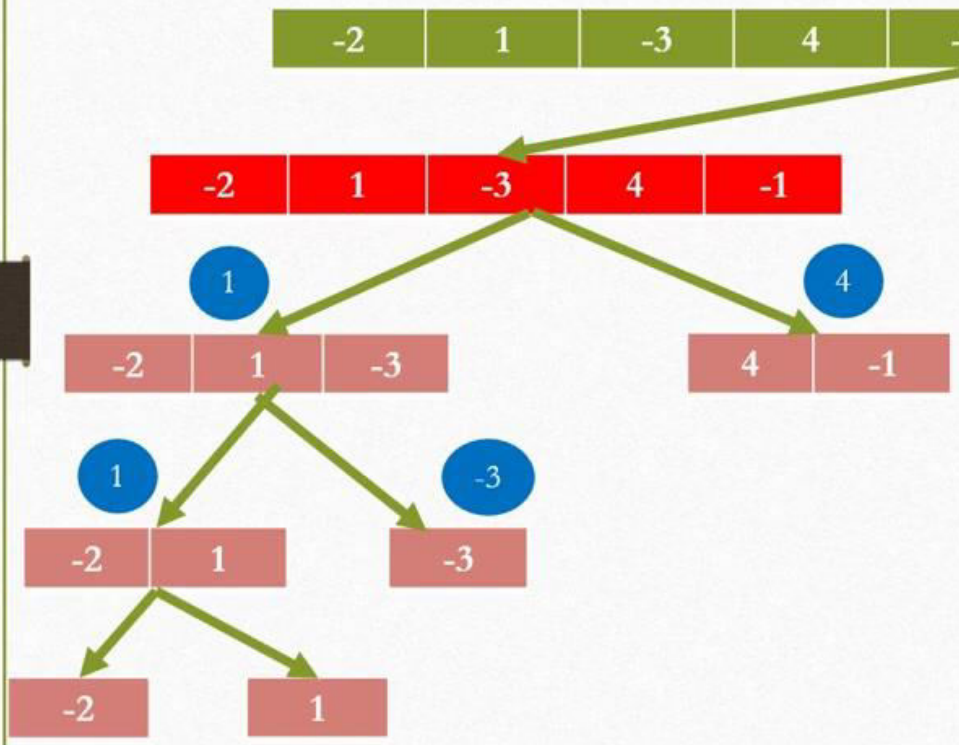
Cross Sum = Left Sub Sum+Right Sub Sum

= (-2+4) = 2

Cross Sum = 2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 4

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -3

Left Sum = 1

Right Sum = 4

Cross Sum Calculations :

Left Sub Sum = **Max(-3, -3+1, -3+1-2) = -2**

Right Sub Sum = **Max(4, 4-1) = 4**

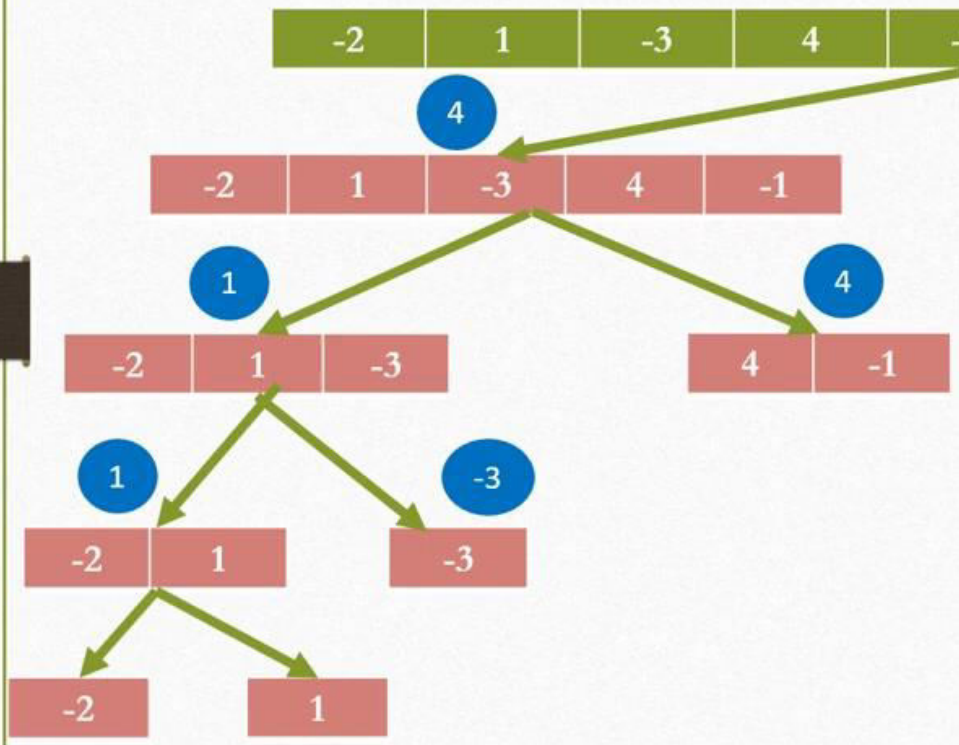
Cross Sum = Left Sub Sum+Right Sub Sum

= (-2+4) = 2

Cross Sum = 2

Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 4

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



Calculations :

Pivot Element: = -3

Left Sum = 1

Right Sum = 4

Cross Sum Calculations :

Left Sub Sum = **Max(-3, -3+1, -3+1-2) = -2**

Right Sub Sum = **Max(4, 4-1) = 4**

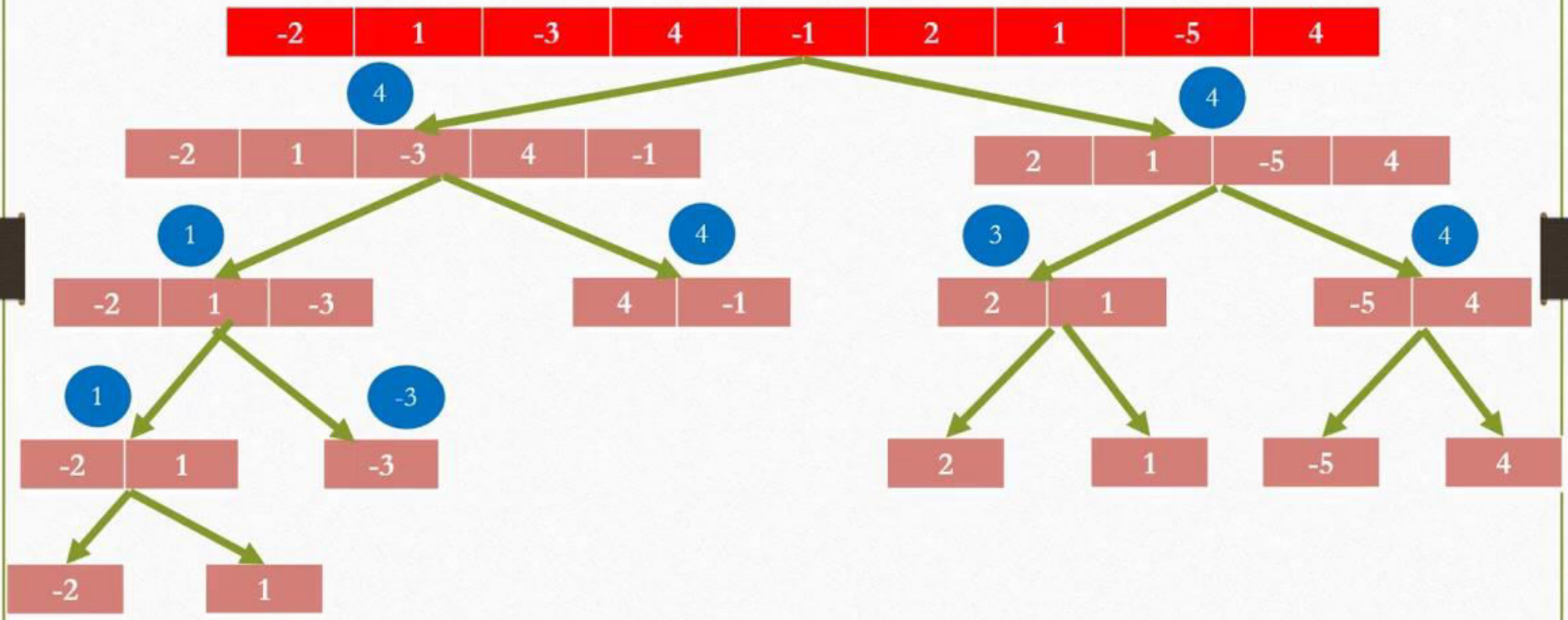
Cross Sum = Left Sub Sum+Right Sub Sum

= (-2+4) = 2

Cross Sum = 2

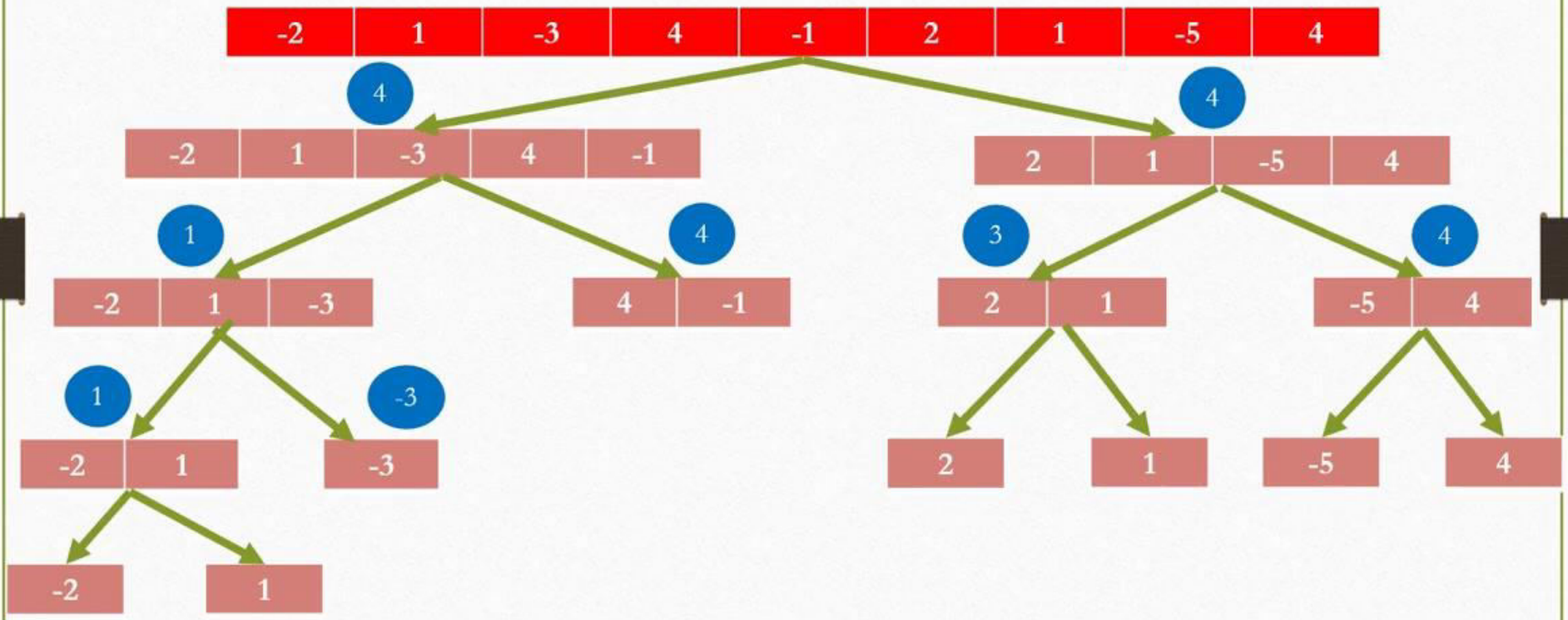
Maximum Sum = Max (Left Sum, Right Sum Cross Sum) = 4

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4



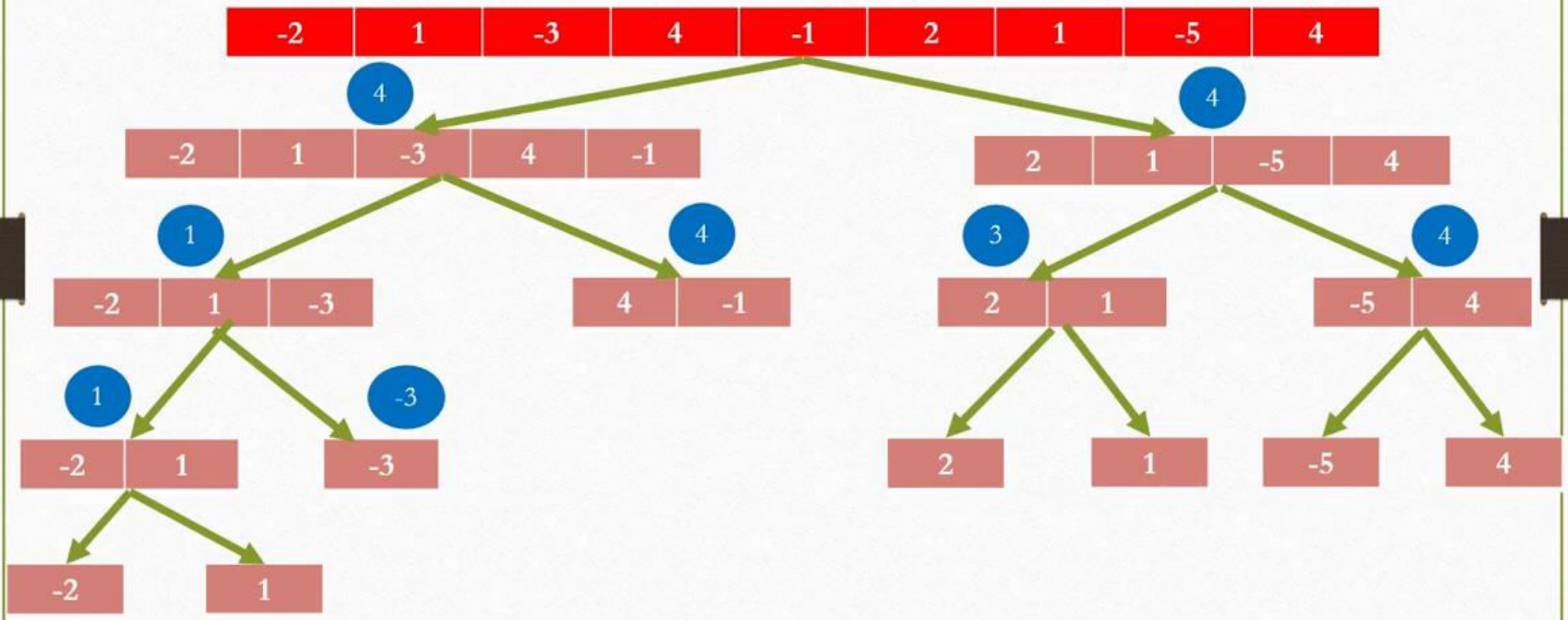
Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4,
Cross Sum = ??



Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4,
Cross Sum = ??



Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4,
Cross Sum = ??

-2	1	-3	4	-1	2	1	-5	4

Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, -1+4, -1+4-3, -1+4-3+1, -1+4-3+1-2)
= Max (-1, 3, 0, 1, -2)
= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)
= Max (2, 3, -2, 2)
= 3

Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

Cross Sum = 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4, Cross Sum = ??

-2	1	-3	4	-1	2	1	-5	4

Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, -1+4, -1+4-3, -1+4-3+1, -1+4-3+1-2)
= Max (-1, 3, 0, 1, -2)
= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)
= Max (2, 3, -2, 2)
= 3

Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

Cross Sum = 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4, Cross Sum = ??

-2	1	-3	4	-1	2	1	-5	4

Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, -1+4, -1+4-3, -1+4-3+1, -1+4-3+1-2)
= Max (-1, 3, 0, 1, -2)
= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)
= Max (2, 3, -2, 2)
= 3

Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

Cross Sum = 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4, Cross Sum = ??

Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, -1+4, -1+4-3, -1+4-3+1, -1+4-3+1-2)
= Max (-1, 3, 0, 1, -2)
= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)
= Max (2, 3, -2, 2)
= 3

Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

Cross Sum = 6

Answer : 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, **-1+4**, -1+4-3, -1+4-3+1, -1+4-3+1-2)

= Max (-1, 3, 0, 1, -2)

= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)

= Max (2, 3, -2, 2)

= 3

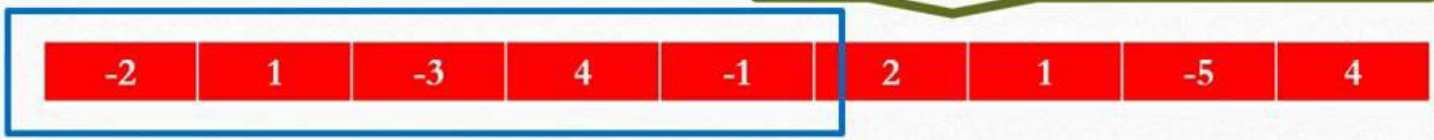
Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

Cross Sum = 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4



Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, -1+4, -1+4-3, -1+4-3+1, **-1+4-3+1-2**)

= Max (-1, 3, 0, 1, -2)

= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)

= Max (2, 3, -2, 2)

= 3

Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

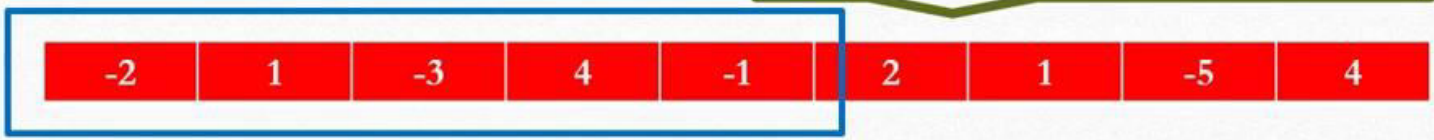
Cross Sum = 6

Answer : 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4



Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, -1+4, -1+4-3, -1+4-3+1, **-1+4-3+1-2**)

= Max (-1, 3, 0, 1, -2)

= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)

= Max (2, 3, -2, 2)

= 3

Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

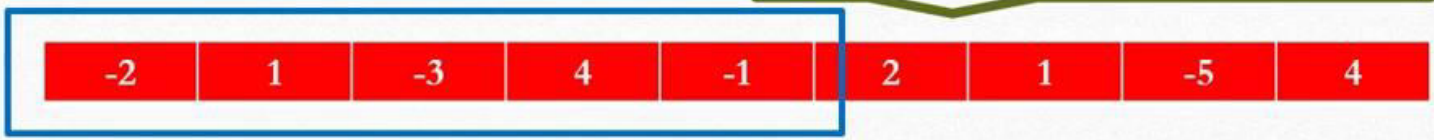
Cross Sum = 6

Answer : 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

Input : -2, 1, -3, 4, -1, 2, 1, -5, 4

Left Sum =4, Right Sum =4



Cross Sum Calculation :

Pivot Element: -1

Left Sub Sum = Max (-1, -1+4, -1+4-3, -1+4-3+1, **-1+4-3+1-2**)

= Max (-1, 3, 0, 1, -2)

= 3

Right Sub Sum = Max (2, 2+1, 2+1-5, 2+1-5+4)

= Max (2, 3, -2, 2)

= 3

Cross Sum = Left Sub Sum + Right Sub Sum = (3+3) = 6

Cross Sum = 6

Answer : 6

Maximum Sum = Max (Left Sum, Right Sum, Cross Sum) = 6

```
FIND-MAXIMUM-SUBARRAY(A, low, high)
 1  if high == low
 2       return (low, high, A[low])          // base case: only one element
 3  else mid = ⌊(low + high)/2⌋
 4       (left-low, left-high, left-sum) =
              FIND-MAXIMUM-SUBARRAY(A, low, mid)
 5       (right-low, right-high, right-sum) =
              FIND-MAXIMUM-SUBARRAY(A, mid + 1, high)
 6       (cross-low, cross-high, cross-sum) =
              FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
 7       if left-sum ≥ right-sum and left-sum ≥ cross-sum
 8            return (left-low, left-high, left-sum)
 9       elseif right-sum ≥ left-sum and right-sum ≥ cross-sum
10            return (right-low, right-high, right-sum)
11       else return (cross-low, cross-high, cross-sum)
```

The running time T(n), recurrence relation of finding maximum sub-array is:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\,, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1\,. \end{cases}$$