

1 Maximum Sum Sub-array Problem

This algorithm finds a sub-array with maximum sum in an array A which starts from `low` and ends at `high` in cubic time.

Algorithm 1 FIND-MAXIMUM-SUBARRAY-CUBIC($A, \text{low}, \text{high}$)

```
1: maxSum  $\leftarrow -\infty$  ▷ Initialize maximum sum
2: startIndex  $\leftarrow \text{low}$  ▷ Start index of the sub-array
3: endIndex  $\leftarrow \text{low}$  ▷ End index of the sub-array
4: for  $i \leftarrow \text{low}$  to  $\text{high}$  do ▷ Fix the start index of the sub-array to  $i$ 
5:   for  $j \leftarrow i$  to  $\text{high}$  do ▷ Fix the end index of the sub-array to  $j$ 
6:     sum  $\leftarrow$  FIND-SUM( $A, i, j$ ) ▷ Find the sum of the sub-array using Algorithm 2
7:     if sum  $>$  maxSum then ▷ Sum of the sub-array is greater than the current maximum sum
8:       maxSum  $\leftarrow$  sum ▷ Update maximum sum
9:       startIndex  $\leftarrow i$  ▷ Update start index of the sub-array
10:      endIndex  $\leftarrow j$  ▷ Update end index of the sub-array
11:     end if
12:   end for
13: end for
14: return (maxSum, startIndex, endIndex)
```

This algorithm computes the sum of a sub-array which starts from `low` and ends at `high` in an array A .

Algorithm 2 FIND-SUM($A, \text{low}, \text{high}$)

```
1: sum  $\leftarrow 0$ 
2: for  $i \leftarrow \text{low}$  to  $\text{high}$  do
3:   sum  $\leftarrow$  sum +  $A[i]$ 
4: end for
5: return sum
```

This algorithm finds a sub-array with maximum sum in an array A which starts from `low` and ends at `high` in quadratic time.

Algorithm 3 FIND-MAXIMUM-SUBARRAY-QUADRATIC($A, \text{low}, \text{high}$)

```
1: maxSum  $\leftarrow -\infty$  ▷ Initialize maximum sum
2: startIndex  $\leftarrow \text{low}$  ▷ Start index of the sub-array
3: endIndex  $\leftarrow \text{low}$  ▷ End index of the sub-array
4: for  $i \leftarrow \text{low}$  to  $\text{high}$  do ▷ Fix the start index of the sub-array to  $i$ 
5:   sum  $\leftarrow 0$  ▷ Initialize the sum of the sub-array starting from index  $i$  to 0
6:   for  $j \leftarrow i$  to  $\text{high}$  do ▷ Fix the end index of the sub-array to  $j$ 
7:     sum  $\leftarrow$  sum +  $A[j]$  ▷ Find the sum of the sub-array starting from index  $i$  and ending at index  $j$ 
8:     if sum  $>$  maxSum then ▷ Sum of the sub-array is greater than the current maximum sum
9:       maxSum  $\leftarrow$  sum ▷ Update maximum sum
10:      startIndex  $\leftarrow i$  ▷ Update start index of the sub-array
11:      endIndex  $\leftarrow j$  ▷ Update end index of the sub-array
12:     end if
13:   end for
14: end for
15: return (maxSum, startIndex, endIndex)
```

Algorithm 4 FIND-MAXIMUM-SUBARRAY-LINEAR-LOGARITHM($A, \text{low}, \text{high}$)

```
1: if low = high then
2:   return ⟨low, high, A[low]⟩
3: else
4:   mid ← ⌊(low+high)/2⌋
5:   ⟨left-low, left-high, left-sum⟩ ← FIND-MAXIMUM-SUBARRAY-LINEAR-LOGARITHM( $A, \text{low}, \text{mid}$ )
6:   ⟨right-low, right-high, right-sum⟩ ← FIND-MAXIMUM-SUBARRAY-LINEAR-LOGARITHM( $A, \text{mid} + 1, \text{high}$ )
7:   ⟨cross-low, cross-high, cross-sum⟩ ← FIND-MAXIMUM-CROSSING-SUBARRAY( $A, \text{low}, \text{mid}, \text{high}$ )
8:   if left-sum ≥ right-sum && left-sum ≥ cross-sum then
9:     return ⟨left-low, left-high, left-sum⟩
10:  else if right-sum ≥ left-sum && right-sum ≥ cross-sum then
11:    return ⟨right-low, right-high, right-sum⟩
12:  else
13:    return ⟨cross-low, cross-high, cross-sum⟩
14:  end if
15: end if
```

Algorithm 5 FIND-MAXIMUM-CROSSING-SUBARRAY($A, \text{low}, \text{mid}, \text{high}$)

```
1: left-sum ←  $-\infty$ 
2: sum ← 0
3: for  $i \leftarrow \text{mid}$  downto low do
4:   sum ← sum +  $A[i]$ 
5:   if sum > left-sum then
6:     left-sum ← sum
7:     max-left ←  $i$ 
8:   end if
9: end for
10: right-sum ←  $-\infty$ 
11: sum ← 0
12: for  $i \leftarrow \text{mid} + 1$  downto high do
13:   sum ← sum +  $A[i]$ 
14:   if sum > right-sum then
15:     right-sum ← sum
16:     max-right ←  $i$ 
17:   end if
18: end for
19: return ⟨max-left, max-right, left-sum + right-sum⟩
```

Questions:

- Maximum number of sub-arrays in an array of size n
- What is the time complexity to find the sum of an array with n elements
- Can we have multiple sub-arrays that give the same sum and that sum is maximum
 - How many sub-arrays can give you the maximum sum
- Maximum/minimum size of the sub-array
- Recurrence relation of divide-and-conquer technique to solve Maximum Sum Sub-array Problem
- Time complexity of the Maximum Sum Sub-array Problem when all the elements are positive
- Time complexity of the Maximum Sum Sub-array Problem when all the elements are negative