
CS29003 ALGORITHMS LABORATORY
ASSIGNMENT 3
Date: 01 – August – 2019

Divide and Conquer

Problem Statement

In a galaxy far far away, scientists have figured out a way to evaluate DNA structures accurately. They have developed a unique representation of a DNA as a sequence $A = \langle a_1, a_2, \dots, a_n \rangle$ where $\forall a_i \in A \ a_i \in \mathbb{Z}$. The value of n is usually very large. For their method to work properly, they need to take a segment $S_{l,r} = \langle a_l, a_{l+1}, \dots, a_r \rangle$, $1 \leq l \leq r \leq n$ in A and evaluate it by calculating a score which is defined by $ev(S_{l,r})$. They want to identify a segment $S_{l,r}$ from a given A for which $ev(S_{l,r})$ is maximum among all possible segments of A . The score is calculated as $ev(S_{l,r}) = (\sum_{i=l}^r a_i)$.

Clearly, for a sequence A of size n , the number of all possible segments in A is of $O(n^2)$. So, evaluating all the segments to find the required solution for a typical value of n is computationally expensive, and thus, they want to minimize the calculation $ev()$ as much as possible. Luckily they know that someone with the knowledge of *Divide and Conquer* methods can provide algorithms to find a solution to this problem that has the complexity of order $O(n \log n)$. Your task is to devise such an algorithm for assisting them.

Algorithm Sketch

```
1 int Max_Cross_Seg (...)
2 {
3     // Figure out the logic for calculating the max segment for crossing case
4     // return the sum
5 }

1 int Max_Seg (...)
2 {
3     if base_case // figure out the base case
4     {
5         max_val = ...; // figure out return value
6     }
7     else
8     {
9         left_seg_max = Max_Seg (...); // calculate for left half
10        right_seg_max = Max_Seg (...); // calculate for right half
11        cross_seg_max = Max_Cross_Seg (...); // calculate for the case where max-segment
        crosses the mid
12        max_val = ...; // figure out return value
13    }
14
15    return max_val;
16 }
```

Example

```
1 Example Seq:
2 [3, 5, -2, 4, -54, 2]
3
4 All Segments and their Sum:
5 Seq from index 0 to index 0 :: Segment >> [3] >> 3
6 Seq from index 1 to index 1 :: Segment >> [5] >> 5
7 Seq from index 2 to index 2 :: Segment >> [-2] >> -2
8 Seq from index 3 to index 3 :: Segment >> [4] >> 4
9 Seq from index 4 to index 4 :: Segment >> [-54] >> -54
10 Seq from index 5 to index 5 :: Segment >> [2] >> 2
11 Seq from index 0 to index 1 :: Segment >> [3, 5] >> 8
12 Seq from index 1 to index 2 :: Segment >> [5, -2] >> 3
13 Seq from index 2 to index 3 :: Segment >> [-2, 4] >> 2
14 Seq from index 3 to index 4 :: Segment >> [4, -54] >> -50
15 Seq from index 4 to index 5 :: Segment >> [-54, 2] >> -52
16 Seq from index 0 to index 2 :: Segment >> [3, 5, -2] >> 6
17 Seq from index 1 to index 3 :: Segment >> [5, -2, 4] >> 7
18 Seq from index 2 to index 4 :: Segment >> [-2, 4, -54] >> -52
19 Seq from index 3 to index 5 :: Segment >> [4, -54, 2] >> -48
20 Seq from index 0 to index 3 :: Segment >> [3, 5, -2, 4] >> 10
21 Seq from index 1 to index 4 :: Segment >> [5, -2, 4, -54] >> -47
22 Seq from index 2 to index 5 :: Segment >> [-2, 4, -54, 2] >> -50
23 Seq from index 0 to index 4 :: Segment >> [3, 5, -2, 4, -54] >> -44
24 Seq from index 1 to index 5 :: Segment >> [5, -2, 4, -54, 2] >> -45
25 Seq from index 0 to index 5 :: Segment >> [3, 5, -2, 4, -54, 2] >> -42
26
27 Maximum Segment is : [3, 5, -2, 4] >> 10
```

```
1 Example of cross case for the example sequence :
2 [3, 5, -2, 4, -54, 2]
3
4 Left half >> [3, 5, -2]
5 Max seg >> [3, 5] >> ev() = 8
6
7 Right half >> [4, -54, 2]
8 Max seg >> [4] >> ev() = 4
9
10 The merged list >> [3, 5, -2, 4, -54, 2]
11 Max seg is [3, 5, -2, 4] >> ev() = 10
12 This maximum segment is spawned over both the left and the right half of the list.
```

Submission Guidelines

1. Please do not use any global variable unless you are explicitly instructed so.
2. Please use proper indentation in your code.
3. Please name your file as <roll.no>_A<assignment.no>_P<part.no>. For example, if your roll number is 14CS10001 and you are submitting part 1 of assignment 3, then name your file as 14CS10001_A3_P1.c or 14CS10001_A3_P1.cpp as applicable.
4. Please write your name and roll number at the beginning of your program.