

Problem ID: 12

Project: Maximum Sum Interval

Difficulty: Easy

The maximum subarray problem is the task of finding the contiguous subarray within a one-dimensional array of numbers which has the largest sum. The list usually contains both positive and negative numbers along with 0. Some properties of this problem are:

1. If the array contains all non-positive numbers, then the solution is the number in the array with the smallest magnitude.
2. If the array contains all non-negative numbers, then the problem is trivial and the maximum sum is the sum of all the elements in the list.
3. An empty set not valid.
4. There can be multiple different sub-arrays that achieve the same maximum sum to the problem.

For example, in the array $[-5, 6, 7, 1, 4, -8, 16]$, the maximum sum is 26. That is because adding $6 + 7 + 1 + 4 + -8 + 16$ gives us 26

.

In this project, you have to design a suitable algorithm to find the maximum sum interval for a given array. You can use **Divide and Conquer** or **Dynamic Programming** or **AD Hoc** approaches in this regard. You should not use Brute Force as it may affect the complexity and runtime of the program.

Input:

An array with n number of index.

Output:

The maximum sum interval for the given array.

Marks Distribution

Project Demonstration (Marks: 40%)

- i. Compilation of code without error (Documentation, comment and indentation in the source code must be maintained)
- ii. Testing the project with different set of inputs.
- iii. A set of validation inputs which has been used to test the solution.

Report (Marks: 60%)

1. Problem Statement:
Write a brief statement of the project in general terms according to the problem description provided. Problem statement should not include any example or detailed description.
2. Algorithm Discussion
A detailed discussion about the algorithm or approach to solve the problem in your own language along with pseudo code / step by step flow and examples.
3. Complexity Analysis
A detailed formal discussion about the designed algorithm's runtime and memory complexity.
4. Implementation
Proper commenting of the code to indicate a code block's purpose what it is doing with proper indentation. Focus on important parts of the code.
5. Applications (Bonus)
If you can present any real application scenario that matches with the problem's goal you will get some additional bonus.

N.B: Any kind of extreme plagiarism if detected will be severely penalized.