Program Name – Find the maximum value of arr[j] – arr[i] + arr[l] – arr[k], such that i < j < k < l

Project Category- Dynamic Programming

Programming Paradigm Used- Dynamic Programming

Example-

Let us say our array is – {4, 8, 9, 2, 20}

Then the maximum such value is –> 23 as 9 – 4 + 20 – 2 = 23

Brute Force Method-

We can simply find all the combinations of size 4 and within them permute the negative and positive signs. The maximum value will be the required answer. This method is very inefficient.

Efficient Method (Dynamic Programming)-

Algorithm-

We will use Dynamic Programming to solve this problem. For this we create four 1-Dimensional DP tables.

Let us say there are four DP tables as - table1[] , table2[] , table3[] , table4[]

Then to find the maximum value of arr[j] – arr[i] + arr[l] – arr[k], such that i < j < k < l

1) table1[] will store the maximum value of arr[j]
2) table2[] will store the maximum value of arr[j] – arr[i]
3) table3[] will store the maximum value of arr[j] – arr[i] + arr[l]

4) table4[] will store the maximum value of arr[j] – arr[i] + arr[l] – arr[k]
5) So we iterate through table4[] the to get the maximum value which will be our required answer.

Time Complexity-

O(N), where N is the size of input array

Space Complexity-

Since we are creating four tables to store our values, so space complexity is O(4*N) ~ O(N)

Exercise to the readers-

Find the maximum value of arr[j] – 2*arr[i] + 3*arr[l] – 7*arr[k], such that i < j < k < l