

### Question 9.

Algorithms Assignment 1

**Faaq Bilal**  
**23100104**

## 1 Part 1

We can divide the algorithm into each of its steps, analyze their running time complexity to get bounds for our algorithm's complexity.

The outermost *for* loop will run over all the elements of the array of size  $n$ . This step will have a complexity of  $O(n)$ . The following functions are nested inside this *for* loop, which means that they will be repeated  $n$  times as well.

The nested *for* loop will run for a maximum of  $n - 2$  times, which is still bounded by  $O(n)$ . As noted above, this *for* loop will be repeated  $O(n)$  times as well.

As the final piece of this equation, the integers inside the Array  $A$  have to be summed up. The maximum number of computations for a specific  $B[i, j]$  is equal to the largest difference between  $j$  and  $i$ . This is bounded by  $O(n)$ , as  $i$  ranges  $1 \rightarrow n$  and  $j$  ranges  $i + 1 \rightarrow n$ .

It should be noted that this summation is nested inside the above *for* loop, hence it will be repeated  $O(n^2)$  times.

As each computation is of order  $O(n)$ , and is repeated  $O(n^2)$  times, the overall computational complexity of the algorithm will have an upper bound of  $O(n^3)$ .

## 2 Part 2

We can use the breakdown from Part 1 to help make a point here as well.

The outermost *for* loop will run  $n$  times, the inner *for* loop will run from  $1 \rightarrow n - 2$  times as well.

The innermost summation runs