

# Tarea 6\_7 Problema 3

Los árboles son unas estructuras maravillosas que vienen en distintas formas, las cuales nos permiten ordenar, acceder y almacenar datos de manera eficiente, incluso se pueden analizar muchos problemas y aplicación de algoritmos con el uso de estos. Como varios de sus ayudantes están en Base de Datos se dan cuenta que el tamaño del input o el número de datos total como cota asintótica no es el único parámetro importante, si no que también el número de consultas que se hacen y el tiempo de respuesta de estas también lo es. Por ende en el siguiente ejercicio se le solicitará que dado un arreglo de tamaño  $N$  y un número de consultas  $Q$  que contienen un rango entre  $L$  y  $R$  (que representan los índices en el arreglo) con  $0 \leq L \leq R \leq N$ , retornen la suma del par de números mas grandes que se encuentren entre  $L$  y  $R$ . Eso sí como la velocidad de las consultas tiene que ser eficiente se le impone una cota asintótica de  $O(N + \log(N) * Q)$  en tiempo computacional, por ejemplo usted no se puede demorar  $O(N * Q)$  ya que eso implica que cada consulta toma un tiempo lineal lo cual no es lo esperado, el tiempo esperado para cada consulta corresponde a  $O(\log(N))$ .

Hint: Arme un Segment-Tree con los datos del arreglo y modifíquelo si es necesario.

## Input

La primera línea contiene el número de elemento en el arreglo.

La segunda línea contiene la cantidad de consultas a realizar.

La siguiente línea contiene  $n$  valores corresponden a los elementos del arreglo.

Los siguientes  $Q$  líneas corresponden a cada consulta y cada línea tiene 2 valores que corresponden al intervalo  $L$ - $R$ .

## Output

Imprima una línea en la pantalla con los resultados de las  $Q$  consultas con un espacio en blanco entre cada valor.

## Example

### Input:

```
7
3
13 56 34 75 86 71 90
0 3
1 4
3 6
```

### Output:

```
131 161 176
```