Maximum Sum

You are given an array of integers `a`. A *range sum query* is defined by a pair of non-negative integers `l` and `r` (`l <= r`). The output to a *range sum query* on the given array `a` is the sum of all the elements of `a` that have indices from `l` to `r`, inclusive.

You have the array `a` and a list of *range sum queries* `q`. Find an algorithm that can rearrange the array `a` in such a way that the total sum of all of the query outputs is maximized, and return this total sum.

Example

For `a = [9, 7, 2, 4, 4]` and `q = [[1, 3], [1, 4], [0, 2]]`, the output should be `maximumSum(a, q) = 62`.

You can get this sum if the array `a` is rearranged to be `[2, 9, 7, 4, 4]`. In that case, the first *range sum query* `[1, 3]` returns the sum `9 + 7 + 4 = 20`, the second query `[1, 4]` returns the sum `9 + 7 + 4 + 4 = 24`, and the third query `[0, 2]` returns the sum `2 + 9 + 7 = 18`. The total sum will be `20 + 24 + 18 = 62`.

Input/Output

- **[execution time limit] 4 seconds (js)**
- **[input] array.integer a**

  An initial array.

  Guaranteed constraints:

  $2 \le a.length \le 10$,

  $1 \le a[i] \le 10$.

- **[input] array.array.integer q**

  An array of *range sum queries*, where each query is an array of two non-negative integers.

  Guaranteed constraints:

  $1 \le q.length \le 10$,

  $0 \le q[i][0] \le q[i][1] < a.length$.

- **[output] integer**

  Return the maximum possible total sum of the given *range sum query* outputs.

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
    console.log("This prints to the console when you Run Tests");
    return "Hello, " + name;
}
```