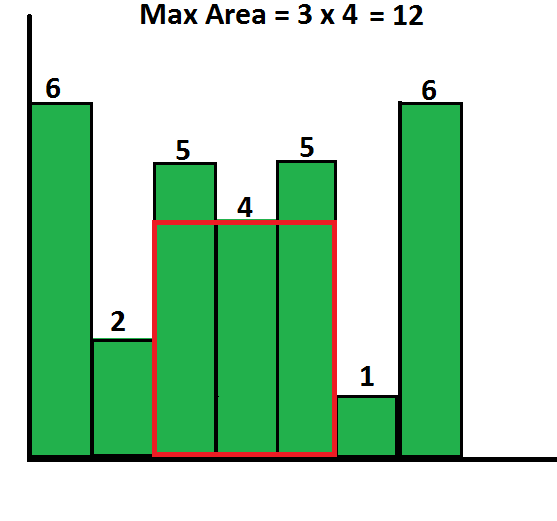
**Area under histogram**

**Problem Statement:**

You will be given an array ***arr***of height of bars, of size ***n***. You have to answer ***q*** queries, where in each query, you will be given left index ***l*** and right index ***r***. For each query, return largest rectangular area possible in a histogram formed using (right-left+1) bars with array of heights as [***arr***[left], ***arr***[left+1], ***arr***[left+2], ..., ***arr***[right]]. Largest rectangle can be made of a number of contiguous bars. For simplicity, you can assume that all bars have same width and the width is 1 unit.

For example, consider the following histogram with 7 bars of heights [6, 2, 5, 4, 5, 1, 6]. The largest possible rectangle possible is 12 (see the below figure, the max area rectangle is highlighted in red).



(source: https://goo.gl/fTcCTK)

**Input/Output Format For The Function:**

**Input Format:**

There are three arguments: ***arr***, denoting input array of height of bars, ***l*** denoting left and ***r*** denoting right as explained in problem statement.

**Output Format:**

Return a number ***maxArea***, denoting maximum rectangular area possible in a histogram formed as explained in problem statement.

**Input/Output Format For The Custom Input:**

**Input Format:**

The first line of the input should contain a single integer ***n,*** denoting the size of input array ***arr***. In the next ***n*** lines, ith line should contain a number ***arr[i]***, denoting ith number of the input array ***arr***, i=(1,2,...,n).

Next line should contain ***q***, denoting no. of queries that need to be answered. In next 2\****q*** lines, (2\*i-1)th line should contain left value for ith query and (2\*i)th line should contain right value for ith query, i=(1,2,...,q), i.e. 1st and 2nd line should contain left and right values for 1st query, 3rd and 4th line should contain left and right values for 2nd query, and so on...

If ***n*** = 5, ***arr***= [2, 4, 6, 5, 8], ***q*** = 2, for 1st query: ***l*** = 0 and ***r***= 4 and for 2nd query: ***l*** = 3 and ***r*** = 3, then input should be:

5

2

4

6

5

8

2

0

4

3

3

**Output Format:**

There will be ***q*** lines, where ith line contains a number ***maxArea[i]***, denoting result of ith query.

For input ***n*** = 5, ***arr***= [2, 4, 6, 5, 8], ***q*** = 2, for 1st query: ***l*** = 0 and ***r***= 4 and for 2nd query: ***l*** = 3 and ***r*** = 3, output will be:

16

5

**Constraints:**

* 1 <= ***n*** <= 2\*10^5
* 1 <= ***q*** <= 10
* 1 <= ***arr***[i] <= 10^9, i=(0,1,2,3,...,n-1)
* 0 <= ***l*** <= ***r***< ***n***for each query.

**Sample Test Cases:**

**Sample Input 1:**

arr = [6, 2, 5, 4, 5, 1, 6]

q = 1

For 1st query: l = 0 and r = 6.

**Sample Output 1:**

12

**Explanation 1:**

1st query: A rectangle of area 12 can be formed using 2nd to 4th bar (0-based indexing) and has maximum area possible in histogram out of all possible rectangles that can be formed using contiguous bar with given array of heights [arr[0],…,arr[6]] = [6, 2, 5, 4, 5, 1, 6] as l=0 and r=6.

**Sample Input 2:**

arr = [2, 4, 6, 5, 8]

q = 2

For 1st query: l = 0 and r = 4.

For 2nd query: l = 3 and r = 3.

**Sample Output 2:**

16

5

**Explanation 2:**

1st query: A rectangle of area 16 can be formed using 1st to 4th bar (0-based indexing) and has maximum area possible in histogram out of all possible rectangles that can be formed using contiguous bar with given array of heights [arr[0], …, arr[4]] = [2, 4, 6, 5, 8] as l=0 and r=4.

2nd query: A rectangle of area 5 can be formed using 3rd to 3rd bar (0-based indexing) and has maximum area possible in histogram out of all possible rectangles that can be formed using contiguous bar with given array of heights [arr[3]] = [5] as l=3 and r=3.

**Suggestions:**

Suggested time in interview: 30 minutes.

The “Suggested Time” is the time expected to complete this question during a real-life interview, not now in homework i.e. For the first attempt of a given homework problem, the focus should be to understand what the problem is asking, what approach you are using, coding it, as well as identifying any gaps that you can discuss during a TA session.

Take your time, but limit yourself to 2 one hour sessions for most problems.