**Problem:**

Given *n* non-negative integers *a1*, *a2*, ..., *an*, where each represents a point at coordinate (*i*, *ai*). *n* vertical lines are drawn such that the two endpoints of line *i* is at (*i*, *ai*) and (*i*, 0). Find two lines, which together with x-axis forms a container, such that the container contains the most water.

**Note:**You may not slant the container and *n* is at least 2.



The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container can contain is 49.

**Solution:**

public class Solution {

public int MaxArea(int[] A) {

int l = 0;

int len = A.Length;

int r = len -1;

int area = 0;

while (l < r)

{

// Calculating the max area

area = Math.Max(area,

Math.Min(A[l], A[r]) \* (r - l));

if (A[l] < A[r])

l += 1;

else

r -= 1;

}

return area;

}

}