CONTAINER WITH MOST WATER

LANGUAGE:JAVA

Given n non-negative integers $a_1, a_2, ..., a_n$, where each represents a point at coordinate (i, a_i). n vertical lines are drawn such that the two endpoints of the line i is at (i, a_i) and $(i, 0)$. Find two lines, which, together with the x-axis forms a container, such that the container contains the most water.

Notice that you may not slant the container.

v

```
Input: height = [1,8,6,2,5,4,8,3,7]
Output: 49
```

IDEAS:
->Here we need to find the maximum area to fill the water in the container.
->area=length*width
-> Given width of the each container is one
->Now task is to find the maximum height to get the maximum area.so width should be max and height should also be max
->If width need's to be maximum then we need to find longer array length.so take two variables where one variable point the lower index of the array and other points to higher index of the array and by using looping condition we iterate through each number in the array and take the minimum value of the pointing indexes height values multiply with (high index-low index) and store it in a variable called area.if the lower index value is lower than higher index value then increase the lower index else decrement higher index.

SOLUTION:

```java
class Solution
{
    public int maxArea(int[] height)
    {
        int low=0;
        int high=height.length-1;
        int area=0;

        while(low<high)
        {

area=Math.max(area,Math.min(height[low],height[high])*(high
-low));
            if(height[low]<height[high])
            {
                low++;
            }
            else
            {
                high--;
            }
        }
        return area;
    }
}
```