

```
1  /*Author: Bochen (mddboc@foxmail.com)
2  Last Modified: Tue Apr 10 22:28:44 CST 2018*/
3
4  /*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.
5
6      Note: You may not slant the container and n is at least 2.*/
7
8
9  import java.lang.System;
10 import java.util.*;
11 import java.lang.Math;
12 import java.util.HashMap;
13
14
15 class ListNode {
16     int val;
17     ListNode next;
18
19     ListNode(int x) {
20         val = x;
21     }
22 }
23
24
25 public class Main {
26     public static void main(String[] args) {
27         int x = 1;
28
29         Solution solution = new Solution();
30
31         boolean receive = solution.isPalindrome(x);
32
33
34         System.out.println("haha");
35     }
36 }
37
38
39 }
40
41
42 class Solution {
43     public int maxArea(int[] height) {
44
45         int startPointer = 0, endPointer = height.length - 1;
46         int maxAreaValue = 0;
47         int startPointerHeight = 0, endPointerHeight = 0;
48
49         while (startPointer < endPointer) {
50
51             startPointerHeight = height[startPointer];
52             endPointerHeight = height[endPointer];
53
54             maxAreaValue = Math.max(maxAreaValue,
55                                     (endPointer - startPointer) * Math.min(startPointerHeight,
56                                     endPointerHeight));
56
57             if ( startPointerHeight <= endPointerHeight ){
58                 startPointer++;
59             }else{
60                 endPointer--;
61             }
62         }
63
64         return maxAreaValue;
65     }
66 }
```