long pressed name

name: A l e x

typed: AA ll eee rr ii

---

name: AA l eee r

typed: AA lll eee rr

---

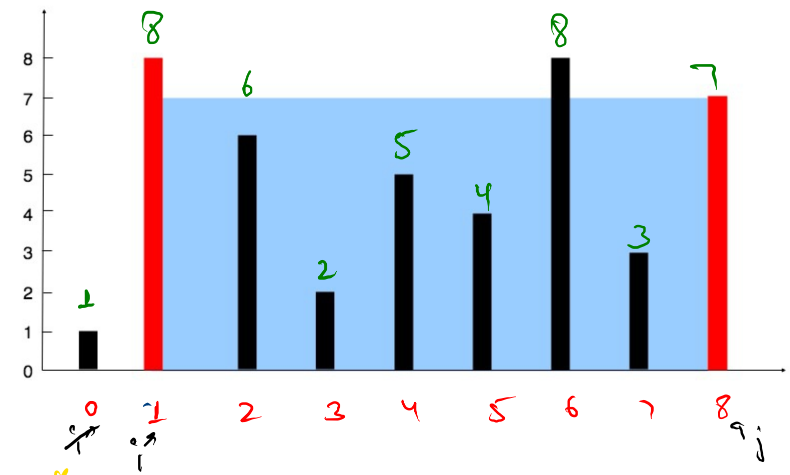
name: AA lll e r

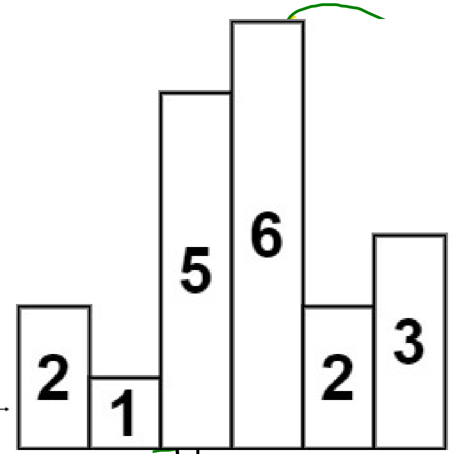
typed: AA ll e r

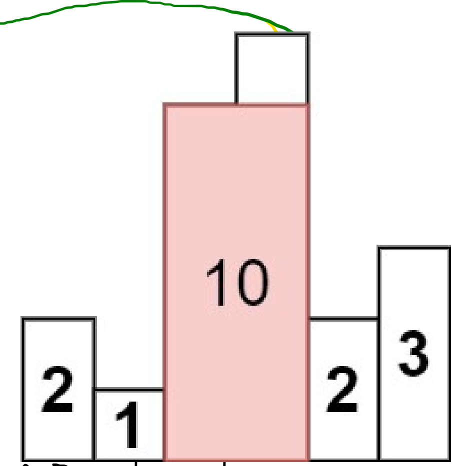
---

name: AA d er

typed: AA l er







0,5 0,6 0,7 0,8

1,6 1,7 1,8

2,7 2,8

arr[0] < arr[4]

Storage: (j − i) * min(arr[i], arr[j])

arr[0] < arr[8]

arr[0] < arr[8]

(i) arr[4] > arr[8] > arr[0]

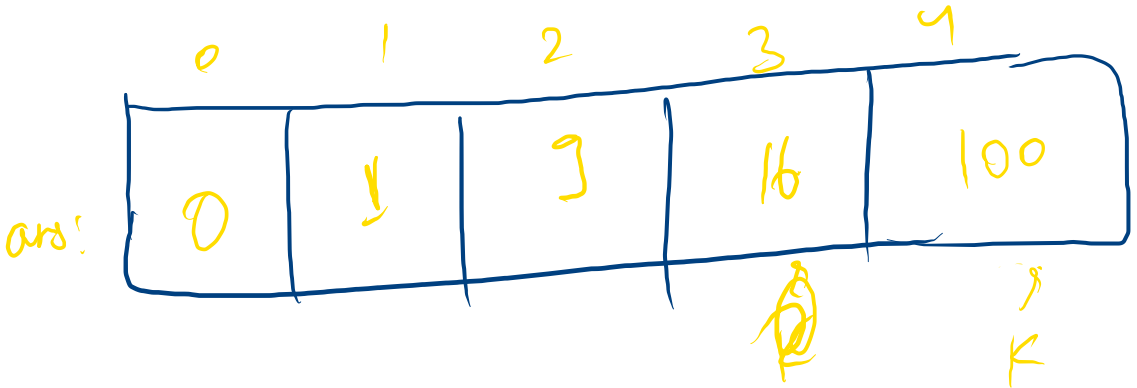
(ii) arr[0] < arr[4] < arr[8]

(iii) arr[4] < arr[0] < arr[8]

```java
class Solution {
    public int maxArea(int[] height) {
        int i =0;
        int j = height.length-1;
        int omax =Integer.MIN_VALUE;

        while(i < j){
            int storage = (j-i) * Math.min(height[i],height[j]);

            omax = Math.max(omax,storage);

            if(height[i] < height[j]){
                i++;
            }else{
                j--;
            }
        }

        return omax;
    }
}
```

Squares

arr: { 0, 1, 4, 9, 16, 25 }



majority element

Boyer moore voting algo

distinct element pairing

2 4 3 2 2 3 2 2 1 2

val = 2 3 2

count = 1 0 1 0 1 2 1 2

(2,4)

(3,2)

(2,3)

(2,1)

```
val = arr[0];
count = 1;
for(int i=1; i<n; i++){
    if(val == arr[i]){
        count++;
    } else if(count==0){
        val = arr[i];
        count=1;
    } else
        count--;
}
```