334. Increasing Triplet Subsequence

Description | Hints | Submissions | Discuss | Solution

Pick One

Given an unsorted array return whether an increasing subsequence of length 3 exists or not in the array.

Formally the function should:

> Return true if there exists *i, j, k*
> such that *arr[i]* < *arr[j]* < *arr[k]* given 0 ≤ *i* < *j* < *k* ≤ *n*-1 else return false.

Note: Your algorithm should run in O(*n*) time complexity and O(*1*) space complexity.

Example 1:

```
Input: [1,2,3,4,5]
Output: true
```

Example 2:

```
Input: [5,4,3,2,1]
Output: false
```

```java
    /*
     * 这个题目只要求找到三个递增的元素即可，不要求这三个元素是否连续，因此，只需维护
     * 两个整形变量a，b，用来记录数组中大小递增的前2个元素，满足条件时，应该有
     * a<b<nums[i];
     */
     public boolean increasingTriplet(int[] nums) {

          int n = nums.length;

          if(n < 3)
              return false;

          int a = Integer.MAX_VALUE;
          int b = Integer.MAX_VALUE;

          for(int i = 0; i < nums.length; i ++) {
              //先判断nums[i]是否小于a，这就意味着a肯定是要小于b的。
              if(nums[i] <= a) {
                  a = nums[i];
              }else if (nums[i] <= b) {
                  b = nums[i];
            }else {
                return true;
            }
          }
         return false;
     }
}
```