

## 递增的三元子序列

给你一个整数数组 `nums`，判断这个数组中是否存在长度为 3 的递增子序列。

如果存在这样的三元组下标  $(i, j, k)$  且满足  $i < j < k$ ，使得  $nums[i] < nums[j] < nums[k]$ ，返回 `true`；否则，返回 `false`。

示例 1:

输入: `nums = [1,2,3,4,5]`

输出: `true`

解释: 任何  $i < j < k$  的三元组都满足题意

示例 2:

输入: `nums = [5,4,3,2,1]`

输出: `false`

解释: 不存在满足题意的三元组

示例 3:

输入: `nums = [2,1,5,0,4,6]`

输出: `true`

解释: 三元组  $(3, 4, 5)$  满足题意，因为  $nums[3] == 0 < nums[4] == 4 < nums[5] == 6$

```
class Solution:
    def increasingTriplet(self, nums: List[int]) -> bool:
        ...
        如果 num>second, 则找到了一个递增的三元子序列, 返回 true;

        否则, 如果 num>first, 则将 second 的值更新为 num

        否则, 将 first 的值更新为 num。

        ...
    if (len(nums)<3):
        return False
    first=nums[0]
    second=float('inf')
```

```
for i in range(len(nums)):
    num=nums[i]
    if num>second:
        return True
    elif num>first:
        second=num
    else:
        first=num
return False
```