

```

1  /*
2  给定一个整数数组，判断是否存在重复元素。
3  如果任何值在数组中出现至少两次，函数返回 true。如果数组中每个元素都不相同，则返回 false。
4  示例 1:
5  输入: [1,2,3,1]
6  输出: true
7  示例 2:
8  输入: [1,2,3,4]
9  输出: false
10 示例 3:
11 输入: [1,1,1,3,3,4,3,2,4,2]
12 输出: true
13 来源: 力扣 (LeetCode)
14 链接: https://leetcode-cn.com/problems/contains-duplicate
15 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
16 */

```

分析:

- 方法一:哈希映射统计法
- 方法二:排序判断法,时间复杂度 $T(n \log n + n) \rightarrow O(n \log n)$

方法一:C++_哈希映射统计法

```

1  class solution
2  {
3      public:
4          bool containsDuplicate(vector<int>& nums)
5          {
6              map<int,int> mii;
7              int i = 0;
8              for(i = 0 ; i < nums.size() ; i++)
9              {
10                 mii[nums[i]] = 0;
11             }
12             return mii.size() == nums.size();
13         }
14     };
15     /*
16     执行结果:
17     通过
18     显示详情
19     执行用时 :40 ms, 在所有 cpp 提交中击败了77.81% 的用户
20     内存消耗 :18.1 MB, 在所有 cpp 提交中击败了5.09%的用户
21     */

```

方法二:C++_排序判断法

```

1  class solution

```

```
2 {
3     public:
4         bool containsDuplicate(vector<int>& nums)
5         {
6             sort(nums.begin(),nums.end());
7             for(int i = 1; i < nums.size() ;i++)
8             {
9                 if(nums[i] == nums[i-1])
10                {
11                    return true;
12                }
13            }
14            return false;
15        }
16 };
17 /*
18 执行结果:
19 通过
20 显示详情
21 执行用时 :24 ms, 在所有 cpp 提交中击败了98.87% 的用户
22 内存消耗 :11.2 MB, 在所有 cpp 提交中击败了80.28%的用户
23 */
```