

贪心: 334. 递增的三元子序列

```
1 class Solution {
2 public:
3     bool increasingTriplet(vector<int>& nums) {
4         int n = nums.size();
5         int a = nums[0], b = INT_MAX;
6         for(int i = 1; i < n; i++)
7         {
8             if(nums[i] > b) return true;
9             else if(nums[i] > a) b = nums[i];
10            else a = nums[i];
11        }
12        return false;
13    }
14 };
```

多源 BFS: 1765. 地图中的最高点

```

1  class Solution {
2      int dx[4] = {1, -1, 0, 0};
3      int dy[4] = {0, 0, 1, -1};
4
5  public:
6      vector<vector<int>> highestPeak(vector<vector<int>>& isWater) {
7          int m = isWater.size();
8          int n = isWater[0].size();
9          queue<pair<int, int>> q;
10         vector<vector<int>> map(m, vector<int>(n, -1));
11         for (int i = 0; i < m; i++) {
12             for (int j = 0; j < n; j++) {
13                 if (isWater[i][j]) {
14                     map[i][j] = 0;
15                     q.push({i, j});
16                 }
17             }
18         }
19         while (q.size()) {
20             auto [a, b] = q.front();
21             q.pop();
22             for (int i = 0; i < 4; i++) {
23                 int x = a + dx[i];
24                 int y = b + dy[i];
25                 if (x >= 0 && x < m && y >= 0 && y < n && map[x][y] == -1) {
26                     map[x][y] = map[a][b] + 1;
27                     q.push({x, y});
28                 }
29             }
30         }
31         return map;
32     }
33 };

```