贪心：674. 最长连续递增序列

```cpp
class Solution {
public:
    int findLengthOfLCIS(vector<int>& nums) {
        int n = nums.size();
        int sum = 0;
        for (int i = 0; i < n;) {
            int j = i + 1;
            while (j < n && nums[j] > nums[j - 1]) {
                j++;
            }
            sum = max(sum, j - i);
            i = j;
        }
        return sum;
    }
};
```

多源BFS：1162. 地图分析

```cpp
class Solution {
    int dx[4] = {1, -1, 0, 0};
    int dy[4] = {0, 0, 1, -1};

public:
    int maxDistance(vector<vector<int>>& grid) {
        int max1 = -1;
        int m = grid.size();
        int n = grid[0].size();
        vector<vector<int>> dist(m, vector<int>(n, -1));
        queue<pair<int, int>> q;
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                if (grid[i][j] == 1) {
                    dist[i][j] = 0;
                    q.push({i, j});
                }
            }
        }
        while (q.size()) {
            auto [a, b] = q.front();
            q.pop();
            for (int i = 0; i < 4; i++) {
                int x = a + dx[i];
                int y = b + dy[i];
                if (x >= 0 && x < m && y >= 0 && y < n && dist[x][y] == -1) {
                    dist[x][y] = dist[a][b] + 1;
                    q.push({x,y});
                    max1 = max(max1, dist[x][y]);
                }
            }
        }
        return max1;
    }
};
```