

第八届长春工业大学程序设计竞赛 题解

21 级出题组 饶晟烨

新闻与传播学院 计算机基础教学中心

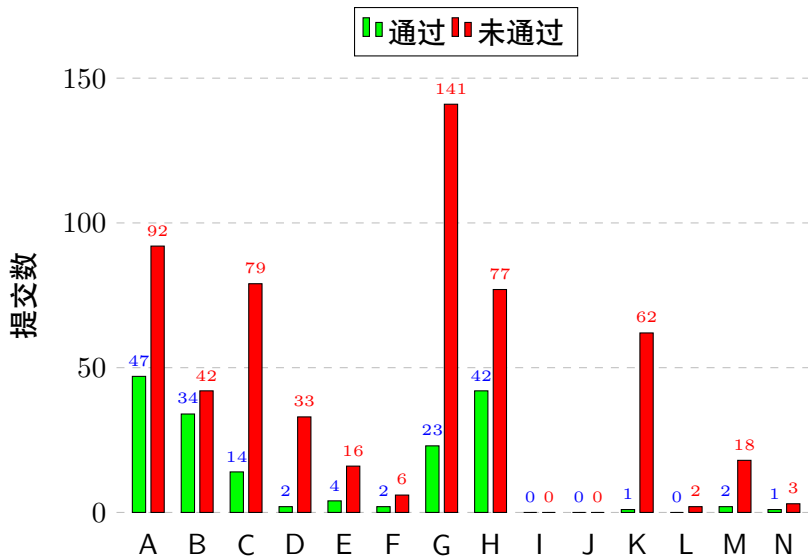
2025 年 3 月 16 日

预期难度

- Easy:A,H,B,G
- Easy-Medium:C,D,
- Medium:E,
- Medium-Hard:
- Hard:K

点击序号可跳转至对应题目。

比赛小结



A. 欢迎参加第八届长春工业大学程序设计竞赛¹

题目大意

- 给定三个整数，判断一个整数与另两个整数平均值的大小关系。
- 关键词：语法题 (选择语句)，签到
- 一血：6min
- 首先先为本题样例出锅道歉。
- 直接计算并判断即可。
- 对 C/C++ 选手：请注意使用浮点型计算及存储平均值。

¹补题链接(提示: 点击标题返回目录页)

```
1  #include <iostream>
2
3  int main()
4  {
5      int a, b, c;
6      std::cin >> a >> b >> c;
7      a *= 2;
8      b += c;
9      if (a < b)
10     {
11         std::cout << "....";
12     }
13     else if (a > b)
14     {
15         std::cout << "emmm";
16     }
17     else
18     {
19         std::cout << "OHHH";
20     }
21 }
22
```

题目大意

- 给定不同阶段的升级所需经验，求 $0 - K$ 的所需经验。
- 关键词：语法题 (选择语句)，签到
- 一血：70min
- 模拟即可。
- 对 C/C++ 选手：请注意开 long long。

```
1  #include <iostream>
2
3  typedef unsigned long long ULL;
4
5  int main()
6  {
7      std::ios::sync_with_stdio(false);
8      ULL x;
9      std::cin >> x;
10     if (x <= 16) std::cout << (x * (x + 6));
11     else if (x <= 31)
12     {
13         ULL diff = x - 16;
14         ULL exp = ((16 + (x - 1)) * diff) / 2;
15         exp *= 5;
16         exp -= 38 * diff;
17         exp += 352;
18         std::cout << exp;
19     }
20     else
21     {
22         ULL diff = x - 31;
23         ULL exp = ((31 + (x - 1)) * diff) / 2;
24         exp *= 9;
25         exp += 1507;
26         exp -= 158 * diff;
27         std::cout << exp;
28     }
29 }
```

B. 祝各位都能取得好成绩³

题目大意

- 给定若干个单词，统计每个字母出现的次数。
- 关键词：语法题 (数组)，签到
- 一血：7min
- 将 26 个英文字母映射到数组下标并统计，输出数组即可。


```
1  #include <iostream>
2  #include <string>
3
4  int count[26];
5
6  int main()
7  {
8      std::ios::sync_with_stdio(false);
9      int n;
10     std::cin >> n;
11     for (int i = 0; i < n; i++)
12     {
13         std::string str;
14         std::cin >> str;
15         for (unsigned long long j = 0; j < str.size(); j++)
16         {
17             count[str[j] - 'a']++;
18         }
19     }
20     for (int i = 0; i < 26; i++)
21     {
22         if (i) std::cout << '\n';
23         std::cout << count[i];
24     }
25 }
```

题目大意

- 累加变量将顺序累加序列，要求任意时刻变量均大于 0，求其初值。
 - 关键词：前缀和，签到
 - 一血：97min
-
- 顺序累加具有前缀性，因此使用前缀和计算每个状态。
 - 若序列均为正数，答案为 1。
 - 若序列存在负数，则答案为前缀和中最小值的相反数 +1。
 - 对 C/C++ 选手：请注意开 long long。



```
1  #include <iostream>
2
3  int main()
4  {
5      std::ios::sync_with_stdio(false);
6      int n;
7      std::cin >> n;
8      long long min = 0;
9      long long sum = 0;
10     for (int i = 0; i < n; i++)
11     {
12         long long x;
13         std::cin >> x;
14         sum += x;
15         min = std::min(min, sum);
16     }
17     std::cout << (-min) + 1;
18 }
```

题目大意

- 在二维网格上有若干个数字 14，其将向上下左右方向上扩散，每扩散一次数字 -1，当相遇时取较大值。输出扩散后的网格。
- 关键词：找规律/BFS
- 一血：53min

本题为 BFS+ 简单剪枝模板题，不再赘述。下面介绍找规律方法：

- 观察数字 14 所在行，发现其为 1, 2, ..., 13, 14, 13, ..., 2, 1。
- 观察 14 所在的上一行和下一行，均为 1, 2, ..., 12, 13, 12, ..., 2, 1。
- 以此类推，可发现本题实质为经典语法题“打印沙漏”变种。

std(BFS+ 剪枝)

```
1 #include <iostream>
2 #include <queue>
3 #include <algorithm>
4
5 class Coordinate
6 {
7 public:
8     int X;
9     int Y;
10    int L;
11    Coordinate(int x, int y, int l): X(x), Y(y), L(l)
12    {
13    }
14 };
15
16 int map[257][257];
17 int vis[257][257];
18 int dir[4][2] =
19 {
20     0, 1,
21     1, 0,
22     0, -1,
23     -1, 0
24 };
25
26 void solution(std::istream &cin, std::ostream &cout)
27 {
28     int n, m, k;
29     cin >> n >> m >> k;
30     for (int kidx = 0; kidx++ < k;)
31     {
32         int x, y;
33         cin >> x >> y;
34         std::queue<Coordinate> que;
35         que.emplace(x, y, 1);
36         vis[x][y] = kidx;
37         while (!que.empty())
38         {
39             Coordinate co = que.front();
40             que.pop();
41             map[co.X][co.Y] = std::max(map[co.X][co.Y], co.L);
42             if (co.L > 1)
43             {
44                 for (int didx[2] : dir)
45                 {
46                     int sx = co.X + didx[0];
47                     int sy = co.Y + didx[1];
48                     if ((sx >= 1) && (sx <= n) && (sy >= 1) && (sy <= m) && (vis[sx][sy] != kidx))
49                     {
50                         vis[sx][sy] = kidx;
51                         que.emplace(sx, sy, co.L + 1);
52                     }
53                 }
54             }
55         }
56     }
57     for (int i = 1; i <= n; i++)
58     {
59         for (int j = 1; j <= m; j++)
60         {
61             cout << map[i][j] << " ";
62         }
63         cout << '\n';
64     }
65 }
66
67 int main()
68 {
69     solution(std::cin, std::cout);
70     std::ios::sync_with_stdio(false);
71 }
```

选手代码 (“打印沙漏”)

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int n,m,k;
4  bool check(int y,int x){
5      return y>0&& y<=n&& x>0&& x<=m;
6  }
7  int main(){
8      cin>>n>>m>>k;
9      vector<vector<int>>>v(n+1,vector<int>(m+1));
10     while(k--){
11         int y,x;cin>>y>>x;
12         for(int i=0;i<=13;i++){
13             for(int j=0;j<=14-i;j++){
14                 if(check(y-i,x+j)){
15                     v[y-i][x+j]=max(v[y-i][x+j],14-i-j);
16                 }
17                 if(check(y-i,x-j)){
18                     v[y-i][x-j]=max(v[y-i][x-j],14-i-j);
19                 }
20                 if(check(y+i,x+j)){
21                     v[y+i][x+j]=max(v[y+i][x+j],14-i-j);
22                 }
23                 if(check(y+i,x-j)){
24                     v[y+i][x-j]=max(v[y+i][x-j],14-i-j);
25                 }
26             }
27         }
28     }
29     for(int i=1;i<=n;i++){
30         for(int j=1;j<=m;j++){
31             cout<<v[i][j]<<' ';
32         }
33         cout<<endl;
34     }
35     return 0;
36 }
```