链滴

SDWU 2022 ACM Individual Training Match 1st 题解

作者：Shun2002

原文链接：https://ld246.com/article/1651071257414

来源网站：链滴

比赛：SDWU 2022 ACM Individual Training Match 1st(已结束)

最终比赛榜单：SDWU 2022 ACM Individual Training Match 1st比赛榜单

比赛题目练习：SDWU 2022 ACM Individual Training Match 1st训练

每场比赛结束后都会创建该比赛的相关练习，点击上方《比赛题目练习》的链接后即可进入，请使用前下发的个人账号进行练习！直接点击题解中每题的标题也可直达练习页面！

请注意！！！提交时编译语言请选择：GNU G++17 7.3.0

G++同时兼容C++与C语言，在正式比赛中也尽量选择高版本编译器

因远程评测题目网络问题，提交后可能有较长时间的 Pending 时间，请耐心等待，若出现 Submitted Failed 问题，请点击刷新符号重新进行提交！

Run ID	题目	状态	分数	运行时间	运行内存	代码长度	语言	判题机
208	GYM-103488L Lexicogr...	Pending	--	--	--	384B	GNU G++17 7.3.0	--
207	GYM-103488J Jiubei an...	Accepted	--	15ms	0 KB	1257B	GNU G++17 7.3.0	judger-alone
206	GYM-103488H Hile and...	Submitted Failed	--	--	--	434B	GNU G++17 7.3.0	--
205	GYM-103488G Generat...	Accepted	--	233ms	0 KB	881B	GNU G++17 7.3.0	judger-alone
204	GYM-103488F Future Vi...	Accepted	--	187ms	100 KB	1731B	GNU G++17 7.3.0	judger-alone
203	GYM-103488E Equality	Submitted Failed	--	--	--	1502B	GNU G++17 7.3.0	--
202	GYM-103488D Disease...	Submitted Failed	--	--	--	534B	GNU G++17 7.3.0	--
201	GYM-103488C Construc...	Submitted Failed	--	--	--	411B	GNU G++17 7.3.0	--
200	GYM-103488B Boboge ...	Accepted	--	15ms	0 KB	254B	GNU G++17 7.3.0	judger-alone
199	GYM-103488A All in!	Pending	--	--	--	96B	GNU G++17 7.3.0	--

原文链接：SDWU 2022 ACM Individual Training Match 1st 题解

A - All in!

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    puts("All in!");
    return 0;
}
```

B - Boboge and Tall Building

```
#include <bits/stdc++.h>
using namespace std;
int T, n, m, k;
int main() {
    scanf("%d", &T);
    while (T -- ) {
        scanf("%d%d%d", &n, &m, &k);
        double h = k * 1.0 / m;
        printf("%.10lf\n", h * (n - 1) * 1.0);
    }
    return 0;
}
```

C - Constructive Problem

打表并找规律

```
//打表
#include <bits/stdc++.h>
using namespace std;
const int N = 10;
int n;
int st[N];
int main() {
    scanf("%d", &n);
    function<void(int, int)> dfs = [&](int d, int sum) {
        if (sum > n) return;
        if (d == n) {
            if (sum != n)   return;
            vector<int> cnt(20);
            for (int i = 0; i < n; i ++ )
                cnt[st[i]] ++;
            bool flag = 1;
            for (int i = 0; i < n; i ++ ) {
                if (cnt[i] != st[i]) {
                    flag = 0;
                    break;
                }
            }
            if (flag) {
                for (int i = 0; i < n; i ++ )
                    printf("%d ", st[i]);
```

```
                puts("");
            }
            return;
        }
        for (int i = 0; i < n; i ++ ) {
            st[d] = i;
            dfs(d + 1, sum + i);
        }
    };
    dfs(0, 0);
    return 0;
}

//提交代码
#include <bits/stdc++.h>
using namespace std;
int n;
int main() {
    scanf("%d", &n);
    if (n < 4 || n == 6)    puts("-1");
    else if (n == 4)    puts("1 2 1 0");
    else if (n == 5)    puts("2 1 2 0 0");
    else {
        printf("%d 2 1 ", n - 4);
        for (int i = 1; i <= n - 7; i ++ )
            cout << 0 << " ";
        cout << 1;
        cout << " 0 0 0" << endl;
    }
    return 0;
}
```

D - Diseased String

```
#include <bits/stdc++.h>
using namespace std;
int T, n, m, ans;
char s[110];
int main() {
    scanf("%d", &T);
    while (T -- ) {
        ans = 0;
        scanf("%d", &n);
        scanf("%s", s + 1);
        for (int i = 1; i <= n; ) {
            if (s[i] != 'y') {
                i ++;
                continue;
            }
            int j = i + 1;
            while (j  <= n && s[j] == 'b')  j ++;
            if (j - i >= 3) ans += j - i - 2;
            i = j;
        }
```

```
        printf("%d\n", ans);
    }
    return 0;
}
```

E - Equality

贪心

特判全部相等以及 m=1 的情况

首先将数组转换为01数组，1表示当前数为最小值，0表示当前数不为最小值

找到第一个1的位置，因为要使得前面的所有数都为1，只能通过第一个1的位置向前拓展

之后从消除完第一个1的位置的前面所有0所得到的最远的起始位置

通过前缀和来快速判断一个区间中是否含有1

如果当前区间含有1，那么一次可以向后拓展m个位置，否则只能拓展m−1个位置

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e5 + 10;
int T, n, m, minn;
int a[N], s[N];
int main() {
    scanf("%d", &T);
    while (T -- ) {
        minn = 0x3f3f3f3f;
        scanf("%d%d", &n, &m);
        for (int i = 1; i <= n; i ++ ) {
            scanf("%d", &a[i]);
            minn = min(minn, a[i]);
        }
        bool flag = 1;
        for (int i = 1; i <= n; i ++ )
            if (a[i] != minn)
                flag = 0;
        if (flag)   puts("0");
        else if (m == 1)    puts("-1");
        else {
            int pos = 0;
            int cnt = 0;
            memset(s, 0, sizeof s);
            for (int i = 1; i <= n; i ++ ) {
                if (a[i] == minn) {
                    pos = i;
                    break;
                }
            }
            int k = (int)ceil((pos - 1) * 1.0 / (m - 1));
            cnt += k;
            int last = max(pos, (m - 1) * k + 1);
            for (int i = 1; i <= last; i ++ )   a[i] = minn;
            for (int i = 1; i <= n; i ++ )
                s[i] = s[i - 1] + (a[i] == minn ? 1 : 0);
```

```
            for (int i = last; i <= n; ) {
                if (a[i] == minn) {
                    i ++;
                    continue;
                }
                else {
                    cnt ++;
                    if (s[min(n, i + m - 1)] - s[i - 1] > 0) i = i + m;
                    else i = i + m - 1;
                }
            }
            printf("%d\n", cnt);
        }
    }
    return 0;
}
```

F - Future Vision

BFS

```
#include <bits/stdc++.h>
using namespace std;
const int N = 110;
int T, n, m;
int d[N][N];
bool st[N][N];
char s[N], g[N][N];
int dx[] = {0, 1, 0, -1};
int dy[] = {1, 0, -1, 0};
struct node {
    int x, y;
    int step;
};
int main() {
    scanf("%d", &T);
    while (T -- ) {
        memset(st, 0, sizeof st);
        memset(d, 0x3f, sizeof d);
        scanf("%d%d", &n, &m);
        int x, y;
        for (int i = 1; i <= n; i ++ ) {
            scanf("%s", s + 1);
            for (int j = 1; j <= m; j ++ ) {
                g[i][j] = s[j];
                if (g[i][j] == 'H')
                    x = i, y = j;
            }
        }
        function<void()> bfs = [&]() {
            queue<node> q;
            q.push({x, y, 0});
            d[x][y] = 0;
            st[x][y] = true;
```

```
        while (q.size()) {
            auto t = q.front();
            q.pop();
            for (int i = 0; i < 4; i ++ ) {
                int tx = t.x + dx[i];
                int ty = t.y + dy[i];
                if (tx < 1 || tx > n || ty < 1 || ty > m || st[tx][ty]|| g[tx][ty] == '#') continue;
                d[tx][ty] = t.step + 1;
                st[tx][ty] = 1;
                q.push({tx, ty, t.step + 1});
            }
        }
    };
    bfs();
    int time, ans;
    bool flag = 0;
    scanf("%d", &time);
    for (int t = 0; t < time; t ++ ) {
        int a, b;
        scanf("%d%d", &a, &b);
        if (g[a][b] == '#') continue;
        if (d[a][b] <= t && !flag) {
            ans = t;
            flag = 1;
        }
    }
    if (flag)   printf("YES %d\n", ans);
    else puts("NO");
  }
  return 0;
}
```

G - Generate 7 Colors

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int N = 10;
int T, n;
int a[N];
signed main() {
  scanf("%lld", &T);
  while (T -- ) {
    for (int i = 0; i < 7; i ++ )
      scanf("%lld", &a[i]);
    bool flag = 1;
    for (int i = 1; i < 7; i ++ )
      if (a[i] > a[i - 1])
        flag = 0;
    if (flag == 0)  {printf("-1\n"); continue;}
    int cnt = 0;
    for (int i = 6; i >= 0; i -- ) {
      if (a[i] == 0)  continue;
      if (i != 6) cnt += a[i];
```

```
            else cnt ++;
            for (int j = 0; j <= i; j ++ )
                a[j] -= a[i];
            if (i == 6)
                for (int j = 0; j <= i; j ++)
                    if (a[j])
                        a[j] --;
            if (a[0] == 0)  break;
        }
        printf("%lld\n", cnt);
    }
    return 0;
}
```

H - Hile and Subsequences' MEX

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int MOD = 998244353;
int T, n;
int qmi(int m, int k, int p)
{
    int res = 1 % p, t = m;
    while (k) {
        if (k & 1) res = res * t % p;
        t = t * t % p;
        k >>= 1;
    }
    return res;
}
signed main() {
    scanf("%lld", &T);
    while (T -- ) {
        scanf("%lld", &n);
        printf("%d\n", qmi(2, n, MOD) - 1);
    }
    return 0;
}
```

I - If I Catch You

该题无题解(水平有限)，如果您做出了这道题，欢迎投稿，我会及时更新文档~

J - Jiubei and Codeforces

模拟

```
#include <bits/stdc++.h>
using namespace std;
int T, n, k, x;
struct node {
```

```
    int l, r;
    string s;
}p[20];
int main() {
    p[1].l = 0xcfcfcfcf, p[1].r = 1199, p[1].s = "Newbie";
    p[2].l = 1200, p[2].r = 1399, p[2].s = "Pupil";
    p[3].l = 1400, p[3].r = 1599, p[3].s = "Specialist";
    p[4].l = 1600, p[4].r = 1899, p[4].s = "Expert";
    p[5].l = 1900, p[5].r = 2099, p[5].s = "Candidate master";
    p[6].l = 2100, p[6].r = 2299, p[6].s = "Master";
    p[7].l = 2300, p[7].r = 2399, p[7].s = "International master";
    p[8].l = 2400, p[8].r = 2599, p[8].s = "Grandmaster";
    p[9].l = 2600, p[9].r = 2999, p[9].s = "International grandmaster";
    p[10].l = 3000, p[10].r = 0x3f3f3f3f, p[10].s = "Legendary grandmaster";
    scanf("%d", &T);
    while (T -- ) {
        scanf("%d%d", &n, &k);
        string s1, s2;
        for (int i = 1; i <= n; i ++ ) {
            int rate = k;
            scanf("%d", &x);
            k += x;
            for (int i = 1; i <= 10; i ++ ) {
                if (p[i].l <= k && p[i].r >= k) s1 = p[i].s;
                if (p[i].l <= rate && p[i].r >= rate) s2 = p[i].s;
            }
            if (s1 != s2) cout << s2 << " -> " << s1 << endl;
        }
        cout << s1 << endl;
    }
    return 0;
}
```

K - Klee and Bomb

该题无题解(水平有限)，如果您做出了这道题，欢迎投稿，我会及时更新文档~

L - Lexicographic Order

如果最后一位是a，直接删除最后一位输出s

如果最后一位不是a，最后一位减1，用z填充剩下的位数直到m

```
#include <bits/stdc++.h>
using namespace std;
int n, m;
string s;
int main() {
    cin >> n >> m >> s;
    if (s[s.size() - 1] == 'a')
        s = s.substr(0, s.size() - 1);
    else {
        char c = s[s.size() - 1];
```

```
            c --;
            s = s.substr(0, s.size() - 1);
            s += c;
            while (s.size() < m)
                s += "z";
        }
        cout << s << endl;
        return 0;
    }
```