

简况

[比赛链接](#)

AC 7题, Rank 9th

总结与反思

cmx

lpy

计算几何还需加强(C题)

要选择正确的容器以及判断是否快读(E题)

xsy

构造题做的太慢了, 规律应该能更早看出来, 分类讨论的时候一定要注意全面。

WA的有两发都是因为自己太愚蠢了, $(1, 2)$ 和 $(2, 1)$ 都输出了`$AA$`还有一次没有删调试信息qwq建议`#ifdef`

题解

A. Is Your System Safe?

题意:

给一个日期判断是否在2017.5.21之后。

题解:

签到题, 随便判。

by MountVoom

B. Tiling Dominoes

题意:

给定一个 $n \times m$ 的矩阵，问能否用 1×2 的多米诺骨牌填满，且不存在一条直线可以在不经过骨牌的情况下把矩形分成两个小矩形。

题解:

构造题，大力打表找规律qwq

首先 $n \times m$ 为奇数一定无解，因为一个骨牌要占2个位置。

我默认 $n < m$ 便于讨论，因为 n 和 m 可以互换，输出的时候处理一下即可。

首先是一些特殊情况 $n = 1$ 时当且仅当 $m = 2$ 有解。

当 m 为奇数时需要满足 $m > 5$ 且 $n > 4$ 且 n 是偶数，然后根据打表得到的规律构造。

```
void cons_m_odd() {
    for (int i = 2; i < n; i += 2)
        ans[i][m] = ans[i - 1][m] = ++cnt;
    ans[n][1] = ans[n - 1][1] = ++cnt;
    for (int i = 2; i <= n - 4; i += 2) {
        ans[i][m - 2] = ans[i + 1][m - 2] = ++cnt;
        ans[i][m - 1] = ans[i + 1][m - 1] = ++cnt;
    }
    ans[n - 3][m - 3] = ans[n - 2][m - 3] = ++cnt;
    ans[n - 3][m - 6] = ans[n - 2][m - 6] = ++cnt;
    ans[n - 2][m - 4] = ans[n - 1][m - 4] = ++cnt;
    ans[n - 2][m - 5] = ans[n - 1][m - 5] = ++cnt;
}
```

接下来在此基础上进行改进。

m 为偶数时，需要满足 $n > 4$ 且 $m > 4$ 。需要注意的是 $n = 5, m = 6$ 有解 $n = 6, m = 6$ 无解。

若此时 n 为偶数，则先按照 $(m - 1, n)$ 构造，然后添加一列，这个部分比较简单不再赘述。

若此时 n 为奇数且 $n > 5$ 可以使用刚才 m 为奇数的构造方式。

最后就只剩下 m 为偶数 $n = 5$ 的情况 $n = 5, m = 6$ 的一个解如下：

```
AABBAB
BCCDAB
BABDCC
CABAAB
CDDCCB
```

接下来不断改造右下角并一直扩展即可，比如 $n = 5, m = 8$ 时解如下：

AABBABEE
 BCCDABFF
 BABDCCGG
 CABAAHHJ
 CDDCCIIJ

by MountVoom

C. Skating on Weiming Lake(补题)

题意:

圆中两个点,找个圆在圆内且不含两点

题解

首先考虑单点,即考虑找一个最大的圆不包含一个点

易知为该点 P 与圆心 C 构成的射线 $\overrightarrow{PC}$ 与圆交于 F ,以 PF 为半径的圆 Γ

判断另一点 Q 是否在 Γ 中即可

```
ans = P + (C - P).unit() * (C.len(P) + R) * 0.5;
if (dcmp(ans.len(P), ans.len(Q)) <= 0) {
    flag = true;
    ansR = ans.len(P);
}
```

而如果 P, Q 两点单点构造出的圆都不满足,考虑由两点一起进行构造

分析可得圆心 D 应该在 PQ 垂直平分线上,且与圆 C 相切于 G

$r = DP = DQ = DG, DG = R - r, DP + DG = R$, 设 $f(D) = DP + DG$ 在垂直平分线上应该是至多存在两点 $D_1, D_2, s.t. \sqcup f(D_i) = R$

且 $f(D)$ 是单峰的

考虑对 $f(D)$ 进行三分,找到峰值 D_0 (为最小值),然后二分得到 D_i

```
Point E = (P + Q) * 0.5;
Vector vertical = (P - Q).unit().rotate();
// E.print();
// vertical.print();
double l = -4e9, r = 4e9;
while (dcmp(l, r) == -1) {
    double a, b;
    a = (2 * l + r) / 3, b = (l + 2 * r) / 3;
    Point A, B;
    A = E + vertical * a, B = E + vertical * b;
    if (dcmp(A.len(P) + A.len(C), B.len(P) + B.len(C)) > 0)
        l = a;
```

```
        else
            r = b;
    }
    double extrempoint = (l + r) / 2;
    // Point X = E + vertical * extrempoint;
    // X.print();
    // printf("%.8lf\n", X.len(P) + X.len(C));
    l = extrempoint, r = 4e9;
    while (dcmp(l, r) == -1) {
        double mid = (l + r) / 2;
        Point A = E + vertical * mid;
        if (dcmp(A.len(P) + A.len(C), R) > 0)
            r = mid;
        else
            l = mid;
    }
    Point tmpans = E + vertical * ((l + r) / 2);

    l = -4e9, r = extrempoint;
    while (dcmp(l, r) == -1) {
        double mid = (l + r) / 2;
        Point A = E + vertical * mid;
        // A.print();
        if (dcmp(A.len(P) + A.len(C), R) > 0)
            l = mid;
        else
            r = mid;
    }
    ans = E + vertical * ((l + r) / 2);
    if (dcmp(ans.len(P), R) == 1)
        ans = tmpans;
    else if (dcmp(tmpans.len(P), R) <= 0 &&
            dcmp(ans.len(P), tmpans.len(P)) <= 0)
        ans = tmpans;
    ansR = ans.len(P);
```

特别注意预处理等于圆心的点

```
Point P(x, y);
if (P == C) P = P + Point(eps, -eps);
```

by Hardict

D. Travel to Qomolangma

E. Walking Around the Country

题意:

有一个完全有向图,给定许多条有向边(会重复),求一条最短路径要求每条边至少经过一次

题解:

很像欧拉通路,不同的是不同块之间不联通,且每个点度数可能不符合要求(出度等于入度)

考虑引入结点0,作为转换结点,用于连接联通块和平衡度数

```
Q[0].clear();
for (int i = 1; i <= N; i++) {
    if (indeg[i] == outdeg[i]) continue;
    int cnt = abs(outdeg[i] - indeg[i]);
    if (outdeg[i] > indeg[i])
        while (cnt--) Q[0].push_back(i);
    else
        while (cnt--) Q[i].push_back(0);
}
```

接着从0点开始寻找欧拉通路即可(注意处理无关结点——没有边相连)

```
void dfs(int u) {
    vis[u] = true;
    while (!Q[u].empty()) {
        int v = Q[u].front();
        Q[u].pop_front();
        dfs(v);
    }
    S.push(u);
}
```

by Hardict

F. A Simple Math Problem

题意:

求解 $(7+4\sqrt{3})^n$ 的整数部分

题解:

$(7+4\sqrt{3})^n = x+y\sqrt{3}$, $(7-4\sqrt{3})^n = x-y\sqrt{3}$

$0 < (7-4\sqrt{3})^n < 1$, $0 < (7+4\sqrt{3})^n < 1 \Rightarrow x, y\sqrt{3}$ 整数部分相差不超过1
又 $x \in \mathbb{N}^+$, $y\sqrt{3} = x-1$, $(7+4\sqrt{3})^n = 2x-1$

在 $\mathbb{Z}(\sqrt{3})$ 空间上快速幂即可

by Hardict

G. Go! Go! Go!

H. Reverse K-th Problem

I. Stairs of Tetris

J. Pairs

K. Lying Island

补题

From:
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:alchemist:pku_campus_2017



Last update: **2020/05/08 22:40**