

memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memseto**
Proudly powered by **Typecho**

BZOJ5403 marshland

2019-01-10 | 题解

这个数据范围容易让人想到网络流，问题在于这个网络流怎么流？

首先可以发现，如果我们要放石头，一定只会放在 $x + y$ 为奇数的格子，否则放了不如不放。所以把这些点拆点，连一条费用为 $a[x][y]$ ，流量为 1 的边。

接下来，我们可以构造一种对 $x + y$ 为偶数的格子的黑白染色方案，使得一个 L 型石头的两个 $x + y$ 为偶数的格子异色。容易发现，我们只需要对所有 x 和 y 都是奇数的格子染黑， x 和 y 都是偶数的格子染白。同理，拆成两个点，中间连一条费用为 0，流量为 1 的边。

同时，为了使得总放置的石头数为 m ，我们还需要再建一个点 P 。

于是源点 S 向 P 连边， P 向黑点连边，黑点向 $x + y$ 为奇数的点连边， $x + y$ 为奇数的点向白点连边，白点向汇点 E 连边即可。

需要注意的是，我们跑最大费用最大流，但是我们只需要费用最大，而不是流量。由于我们采用增广路算法，每次新增的流量的花费一定递减，所以当花费是负数时直接跳出即可。

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

```
// =====
//  author: memset0
//  date: 2019.01.10 08:14:13
//  website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
namespace ringo {
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 55, Nd = 5010, Ed = 1000010;
int n, m, k, l, r, u, s, e, ts, nod, flow, cost;
int a[N][N], id[N][N], id2[N][N], pre[Nd], dis[Nd], q[Nd];
int tot = 2, hed[Nd], nxt[Ed], to[Ed], val[Ed], cst[Ed];
bool b[N][N], inq[Nd];
ll ans;

inline void add_edge(int u, int v, int w, int c) {
    nxt[tot] = hed[u], to[tot] = v, val[tot] = w, cst[tot] = c, hed[u] = tot+
    nxt[tot] = hed[v], to[tot] = u, val[tot] = 0, cst[tot] = -c, hed[v] = tot+
}

bool spfa() {
    memset(pre, 0, sizeof(pre));
    memset(dis, -63, sizeof(dis));
    q[1 = r = 0] = s, dis[s] = 0, inq[s] = 1;
```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

```
while (l <= r) {
    u = q[(l++) % Nd], inq[u] = 0;
    for (int i = hed[u], v = to[i]; i; i = nxt[i], v = to[i])
        if (val[i] && dis[u] + cst[i] > dis[v]) {
            dis[v] = dis[u] + cst[i], pre[v] = i;
            if (!inq[v]) inq[q[(++r) % Nd] = v] = 1;
        }
    }
return pre[e];
}

void main() {
    read(n), read(m), read(k);
    for (int i = 1; i <= n; i++) for (int j = 1; j <= n; j++) read(a[i][j]),
    for (int i = 1, x, y; i <= k; i++) read(x), read(y), b[x][y] = 1;
    s = ++nod, e = ++nod, ts = ++nod;
    add_edge(s, ts, m, 0);
    for (int i = 1; i <= n; i++) for (int j = 1; j <= n; j++) id[i][j] = ++no
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++) {
            static const int nxt[4][2] = {{0, 1}, {1, 0}, {0, -1}, {-1, 0}};
            if (b[i][j]) continue;
            if ((i + j) & 1) {
                add_edge(id[i][j], id2[i][j], 1, a[i][j]);
                for (int t = 0, x, y; t < 4; t++) {
                    x = i + nxt[t][0], y = j + nxt[t][1];
                    if (b[x][y] || x < 1 || y < 1 || x > n || y > n) continue
                    if (x & 1) add_edge(id2[x][y], id[i][j], 1, 0);
                    else add_edge(id2[i][j], id[x][y], 1, 0);
                }
            } else {
                add_edge(id[i][j], id2[i][j], 1, 0);
                if (i & 1) add_edge(ts, id[i][j], 1, 0);
                else add_edge(id2[i][j], e, 1, 0);
            }
        }
    }
while (spfa()) {
```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme Ringo by memseto
Proudly powered by Typecho

BZOJ5403 marshland - memset0 的博客

```
flow = 1e9, cost = 0;
for (int i = pre[e]; i; i = pre[to[i ^ 1]]) flow = std::min(flow, val
for (int i = pre[e]; i; i = pre[to[i ^ 1]]) val[i] -= flow, val[i ^ 1
for (int i = pre[e]; i; i = pre[to[i ^ 1]]) cost += flow * cst[i];
if (cost <= 0) break;
ans -= cost;
}
print(ans, '\n');
}

} signed main() { return ringo::main(), 0; }
```

网络流

用户名

邮箱

网址 (选填)

可以在这里写评论哦 ~

提交评论



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

BZOJ5405 platform

上一篇 «

BZOJ4179 B

» 下一篇

© 2017 - 2019 memset0 的博客.

浙ICP备19006255号-1

97689 visits · 24756 visitors · 74.48 W words

在这里输入关键字哦 ~ (回车搜索)

关于我

友情链接

文章聚合

Theme Ringo by memset0

Proudly powered by Typecho

