

# 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

## 洛谷4102 [HEOI2014]林中路径

2019-01-25 | 题解

数据范围容易想到利用矩阵进行计算。

考虑  $f(k) = (A, B)$ ，其中  $A$  表示恰好等于  $k$  的路径条数， $B$  表示小于等于  $k$  的路径条数。则可以这样转移： $(A, B) \times (C, D) = (A \times C, B + A \times D)$ 。

需要注意的是，对长度为  $L$  的路径对答案的贡献可以表示为一个二次多项式（即  $f(L) = L^2$ ），转移时不能直接计算，而需要记录一次项和常数项系数。注意多项式乘法时需要乘上组合数。

代码：

```
// =====  
//   author: memset0  
//   date: 2019.01.17 17:10:00  
//   website: https://memset0.cn/  
// =====  
#include <bits/stdc++.h>  
#define pb push_back  
#define rep(i, l, r) for (register int i = l; i <= r; i++)  
  
namespace ringo {
```



# 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](#)

```
typedef long long ll;
typedef unsigned long long ull;

inline char read() {
    static const int IN_LEN = 1000000; static char buf[IN_LEN], *s, *t;
    return (s == t ? t = (s = buf) + fread(buf, 1, IN_LEN, stdin), (s == t ?
}
template <class T> inline void read(T &x) {
    static bool f; static char c;
    for (f = 0, c = read(); !isdigit(c); c = read()) { f ^= c == '-'; if (c =
    for (x = 0; isdigit(c); c = read()) x = ((x + (x << 2)) << 1) + (c ^ '0')
    if (f) x = -x;
}
const int OUT_LEN = 10000000; char obuf[OUT_LEN], *ooh = obuf;
inline void print(char c) {
    if (ooh == obuf + OUT_LEN) fwrite(obuf, 1, OUT_LEN, stdout), ooh = obuf;
    *ooh++ = c;
}
template<class T> inline void print(T x) {
    static int buf[30], cnt;
    if (!x) { print('0'); return; }
    if (x < 0) print('-'), x = -x;
    for (cnt = 0; x; x /= 10) buf[++cnt] = x % 10 + '0';
    while (cnt) print((char)buf[cnt--]);
}
inline void flush() { fwrite(obuf, 1, ooh - obuf, stdout); }
template <class T> inline void maxd(T &a, T b) { if (b > a) a = b; }
template <class T> inline void mind(T &a, T b) { if (b < a) a = b; }
template <class T> inline void print(T x, char c) { print(x), print(c); }
template <class T> inline T abs(const T &a) { if (a < 0) return -a; return a;

const int N = 110, mod = 1e9 + 7;
int n, m, k, q;

struct pair {
    struct matrix {
        struct poly {
```



# 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](#)

洛谷4102 [HEOI2014]林中路径 - memseto 的博客

```
int f[3];
inline void inc() { ++f[0], ++f[1], ++f[2]; }
inline void operator += (const poly &other) {
    f[0] += other.f[0]; if (f[0] >= mod) f[0] -= mod;
    f[1] += other.f[1]; if (f[1] >= mod) f[1] -= mod;
    f[2] += other.f[2]; if (f[2] >= mod) f[2] -= mod;
}
friend inline poly operator + (const poly &a, const poly &b) {
    static poly s;
    s.f[0] = a.f[0] + b.f[0]; if (s.f[0] >= mod) s.f[0] -= mod;
    s.f[1] = a.f[1] + b.f[1]; if (s.f[1] >= mod) s.f[1] -= mod;
    s.f[2] = a.f[2] + b.f[2]; if (s.f[2] >= mod) s.f[2] -= mod;
    return s;
}
friend inline poly operator * (const poly &a, const poly &b) {
    static poly s;
    s.f[0] = (ll)a.f[0] * b.f[0] % mod;
    s.f[1] = ((ll)a.f[0] * b.f[1] + (ll)a.f[1] * b.f[0]) % mod;
    s.f[2] = ((ll)a.f[2] * b.f[0] + (ll)a.f[0] * b.f[2] + 2LL * a
    return s;
}
} a[100][100];
inline void inc(int x, int y) { a[x][y].inc(); }
inline int ask(int x, int y) { return a[x][y].f[2]; }
inline void operator += (const matrix &other) {
    for (register int i = 0; i < n; i++)
        for (register int j = 0; j < n; j++)
            a[i][j] += other.a[i][j];
}
friend inline matrix operator + (const matrix &a, const matrix &b) {
    static matrix c;
    for (register int i = 0; i < n; i++)
        for (register int j = 0; j < n; j++)
            c.a[i][j] = a.a[i][j] + b.a[i][j];
    return c;
}
friend inline matrix operator * (const matrix &a, const matrix &b) {
```



# 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

```
static matrix c;
for (register int i = 0; i < n; i++)
    for (register int j = 0; j < n; j++)
        c.a[i][j].f[0] = c.a[i][j].f[1] = c.a[i][j].f[2] = 0;
for (register int i = 0; i < n; i++)
    for (register int k = 0; k < n; k++)
        for (register int j = 0; j < n; j++)
            c.a[i][j] += a.a[i][k] * b.a[k][j];
return c;
}
} A, B;
inline int ask(int x, int y) { return B.ask(x, y); }
inline void inc(int x, int y) { A.inc(x, y), B.inc(x, y); }
inline void operator *= (const pair &other) {
    A = A * other.A, B += A * other.B;
}
friend inline pair operator * (const pair &a, const pair &b) {
    return (pair){a.A * b.A, a.B + a.A * b.B};
}
} a, s;

void main() {
    read(n), read(m), read(k), read(q);
    for (int i = 1, u, v; i <= m; i++) read(u), read(v), --u, --v, a.inc(u, v)
    for (s = a, --k; k >= 1, a = a * a) if (k & 1) s = s * a;
    for (int i = 1, u, v; i <= q; i++) read(u), read(v), --u, --v, print(s.as
}

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

矩阵快速幂

巧妙的思路

用户名



# 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**

邮箱

网址 (选填)

可以在这里写评论哦 ~

提交评论

洛谷5162 WD与积木  
上一篇 «

LOJ2541 「PKUWC2018」猎人杀  
» 下一篇

© 2017 - 2019 **memset0** 的博客.

浙ICP备19006255号-1

97703 visits · 24756 visitors · 74.48 W words



# memset0's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

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**

