20190716解题报告

T1：
首先写了个大模拟，发现根本跑不到50，只能跑到30左右，于是想办法找规律，最后找到规律，愉快做出

T2：
原来的T2，是个错题，于是咕咕咕了
新题换了个二分图匹配，跑个匈牙利算法就过了，板子题
不过，话说这个gg是什么意思？qwq

T3：
一道神奇的需要矩阵快速幂优化的DP题，据gyh大佬说，先跑暴力，然后矩阵优化，最后输出就好了，但我是个菜菜，听不懂啊

代码

T1：

```
#include <iostream>
#include <cmath>
#include <cstdio>
using namespace std;

long long n,ans;
char a;

long long power(long long a, int b) {
    long long res = 1;
    while(b) {
        if(b & 1) res  = res * a;
        a = a * a;
        b >>= 1;
    }
    return res;
}

int main() {
    freopen("ball.in", "r", stdin);
    freopen("ball.out", "w", stdout);
    cin >> n;
    for(int i = 0; i < n; i++) {
        cin >> a;
        if(a == 'B') ans += power(2ll, i);
    }
    cout << ans;
}
```

T2：

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <string>
#define N 1000011
using namespace std;

int n, m, e;
int match[N], head[N], cnt, ans;
bool used[N];

struct node {
    int next, to;
} edge[N];

inline void add(int bg, int ed) {
    edge[++cnt].to = ed;
    edge[cnt].next = head[bg];
    head[bg] = cnt;
}

int find(int x) {
    for(int i = head[x]; i; i = edge[i].next) {
        int u = edge[i].to;
        if(!used[u]) {
            used[u] = 1;
            if(!match[u] || find(match[u])) {
                match[u] = x;
                return 1;
            }
        }
    }
    return 0;
}
```

```
int ocean() {
    freopen("gg.in", "r", stdin);
    freopen("gg.out", "w", stdout);
    scanf("%d%d%d", &n, &m, &e);
    for(int i = 1; i <= e; i++) {
        int u, v;
        scanf("%d%d", &u, &v);
        if(v > m || u > n) continue;
        add(u, v);
    }
    for(int i = 1; i <= n; i++) {
        memset(used, false, sizeof(used));
        if(find(i)) ans++;
    }
    cout << ans << '\n';
    return 0;
}

int loceaner = ocean();
int main() {;}
```

T3：

菜鸡不会T3，于是抄了一遍gyh大佬的代码

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <ctype.h>
#define max(a,b) a>b?a:b
#define min(a,b) a<b?a:b
#define int long long
using namespace std;

inline int read() {
    char c = getchar();
    int x = 0, f = 1;
    for( ; !isdigit(c); c = getchar()) if(c == '-') f = -1;
    for( ; isdigit(c); c = getchar()) x = (x<<3) + (x<<1) + (c^48);
    return x * f;
}

const int mod = 19260817;
int n, m;
struct matrix {
    int map[55][55];
}f, map, ans;
int tmp[2][60];

matrix operator * (const matrix &a, const matrix &b) {
    matrix c;
    memset(c.map, 0, sizeof(c.map));
    for(int k = 1; k <= n; k++) {
        for(int i = 1; i <= n; i++) {
            for(int j = 1; j <= n; j++) {
                c.map[i][j] = (c.map[i][j] + a.map[i][k] * b.map[k][j] % mod
            }
        }
    }
    return c;
}

inline void build(matrix &map) {
    for(int i = 1; i <= n; i++) map.map[i][i] = 1;
}

inline void quick_power(matrix &ans,int k)
{
    while(k)
      {
        if(k&1) ans=ans*map;
        map=map*map;
        k>>=1;
      }
}

signed main() {
    freopen("jump.in", "r", stdin);
    freopen("jump.out", "w", stdout);
    n = read(), m = read();
    int l = n / 2 + 1, r = n / 2 + 1, now = 1;
    tmp[!now][l] = 1;

    for(int i = 2; i <= n; i++) {
        for(int j = l - 1; j <= r + 1; j++) tmp[now][j] = 0;
        for(int j = l; j <= r; j++) {
            tmp[now][j - 1] = (tmp[now][j - 1] + tmp[!now][j]) % mod;
            tmp[now][j] = (tmp[now][j] + tmp[!now][j]) % mod;
            tmp[now][j + 1] = (tmp[now][j + 1] + tmp[!now][j]) % mod;
        }
        now = !now, l = max(l - 1, 1), r = min(r + 1, n);
    }
    if(n >= m) {
        for(int i = 1; i <= n; i++) cout << tmp[!now][i] << ' ';
        return 0;
    }

    for(int i = 1; i <= n; i++) f.map[1][i] = tmp[!now][i];
    for(int i = 1; i <= n; i++) {
        for(int j = i - 1; j <= i + 1; j++) {
            map.map[i][j] = 1;
        }
    }

    build(ans);
    quick_power(ans, m - n);
    f = f * ans;

    for(int i = 1; i <= n; i++) cout << f.map[1][i] << ' ';

    return 0;
}
```

最后，不能忘记Orz GYH DALAO