

# 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**

## LOJ2542 「PKUWC2018」随机游走

2019-01-14 | 题解

考虑 Min-Max 容斥,  $\min(S)$  表示至少一个从  $s$  走到至少一个属于集合  $S$  的点的期望时间。

用  $f(i, S)$  表示集合  $S$  下  $i$  到任意  $u \in S$  的期望时间, 则  $\min(S) = f(s, S)$ 。

$$f(i, S) = \begin{cases} 0 & (i \in S) \\ \frac{\sum_{i \rightarrow j} f(j, S)}{d_i} + 1 & (i \notin S) \end{cases}$$

可通过高斯消元求出, 但复杂度过大不能接受。考虑把原式化为  $f(i) = k_i \times f(\text{father}_i) + b_i$  的形式, 推导过程略。对于每个查询状压处理  $\max(S)$  不现实, 需要处理出每个子集, FWT 一下即可 AC。

代码:

```
// =====  
// author: memset0  
// date: 2019.01.14 10:09:31  
// website: https://memset0.cn/
```



# 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**

```
// =====
#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 = 20, mod = 998244353;
int n, m, t, s, S, lim, f[N], k[N], b[N], a[1 << 18];
std::vector <int> G[N];

inline int sum(int a, int b) { return (a + b) % mod; }
inline int dec(int a, int b) { return (a - b + mod) % mod; }
int inv(int x) { return !x || x == 1 ? 1 : (11)(mod - mod / x) * inv(mod % x)

void dfs(int u, int father = 0) {
    for (auto v : G[u]) if (v != father) dfs(v, u);
    if ((1 << (u - 1)) & S) { k[u] = b[u] = 0; return; }
    int Sk = 0, Sb = 0, d = G[u].size();
    for (auto v : G[u]) if (v != father) (Sk += k[v]) %= mod, (Sb += b[v]) %=
    k[u] = inv(dec(d, Sk)), b[u] = (11)k[u] * sum(Sb, d) % mod;
}
void dfs2(int u, int father = 0) {
    f[u] = ((11)k[u] * f[father] + b[u]) % mod;
    for (auto v : G[u]) if (v != father) dfs2(v, u);
}
```



# 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**

```
int calc(int _S) {
    S = _S, dfs(s), dfs2(s);
    return f[s];
}

void fwt(int *a) {
    for (int len = 1; len < lim; len <= 1)
        for (int i = 0; i < lim; i += (len < 1))
            for (int j = 0; j < len; j++)
                (a[i + j + len] += a[i + j]) %= mod;
}

void main() {
    read(n), read(m), read(s), lim = 1 << n;
    for (int i = 1, u, v; i < n; i++) read(u), read(v), G[u].push_back(v), G[
    for (S = 1; S < lim; S++) a[S] = (1l)calc(S) * (__builtin_popcount(S) & 1
    fwt(a);
    for (int i = 1; i <= m; i++) {
        int S = 0; read(t);
        for (int i = 1, x; i <= t; i++) read(x), S |= 1 << (x - 1);
        print(a[S], '\n');
    }
}

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

巧妙的思路

Min-Max 容斥

高斯消元

用户名

邮箱

网址 (选填)



# memseto's Notebook

方知蓦然回首之时  
那人却已不在灯火阑珊处

可以在这里写评论哦 ~

提交评论

LOJ2541 「PKUWC2018」猎人杀  
上一篇 «

Min-Max 容斥学习笔记  
» 下一篇

关于我  
友情链接  
文章聚合

Theme **Ringo** by **memset0**  
Proudly powered by **Typecho**

© 2017 - 2019 **memset0** 的博客.  
**浙ICP备19006255号-1**  
97705 visits · 24756 visitors · 74.48 W words

在这里输入关键字哦 ~ (回车搜索)



# 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**

