

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

BZOJ5404 party

2019-01-10 | 题解

可能有点假，但没有用线段树.....

首先肯定会在 LCA 处集合，通过树剖 + bitset 维护出可选的集合。然后利用 Hall 定理进行完美匹配统计答案。可以发现直接树剖 + 暴力跳是 $O(n^2)$ 的，所以限制树链的长度至多为 S ，则复杂度为： $O(n + qcS \times \frac{m}{32} + qc \times \frac{n}{S} + q \times 2^c \times \frac{m}{32})$ ，取 $S = \sqrt{n}$ 。

试了一下可以被极端数据卡掉，但是 BZOJ 上的数据是随机生成的所以跑的飞快 QAQ ...

```
// =====
//  author: memset0
//  date: 2019.01.10 14:41:24
//  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();
}
```



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](#)

```

    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 = 3e5 + 10, M = 1001;
int n, m, q, pos, lca, sqn;
int a[N], c[10], top[N], fa[N], dep[N], siz[N], son[N], id[N], wid[N];
std::bitset<M> mix, f[10], pre[N];
std::vector<int> G[N];
const std::bitset<M> empty_bitset;

void dfs(int u) {
    siz[u] = 1;
    if (!G[u].size()) return;
    for (int i = 0, v = G[u][0]; i < (int)G[u].size(); i++, v = G[u][i]) if (
        fa[v] = u, dep[v] = dep[u] + 1, dfs(v), siz[u] += siz[v];
        if (siz[v] > siz[son[u]]) son[u] = v;
    }
}

void dfs(int u, int toppoint, int dep = 1, const std::bitset<M> &from = empty_bitset) {
    top[u] = toppoint, id[u] = ++pos, wid[id[u]] = u;
    pre[u] = from, pre[u].set(a[u]);
    if (siz[u] == 1) return;
    if (dep <= sqn) {
        dfs(son[u], toppoint, dep + 1, pre[u]);
        if (!G[u].size()) return;
        for (int i = 0, v = G[u][0]; i < (int)G[u].size(); i++, v = G[u][i])
            dfs(v, v);
    } else {
        if (!G[u].size()) return;
        for (int i = 0, v = G[u][0]; i < (int)G[u].size(); i++, v = G[u][i])

```



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](#)

```

        dfs(v, v);
    }
}

inline int get_lca(int u, int v) {
    while (top[u] != top[v]) {
        if (dep[top[u]] > dep[top[v]]) std::swap(u, v);
        v = fa[top[v]];
    } return dep[u] < dep[v] ? u : v;
}

inline void get_bitset(int u, int v, std::bitset<M> &f) {
    f.reset();
    while (top[u] != top[v]) f |= pre[u], u = fa[top[u]];
    while (u != v) f.set(a[u]), u = fa[u];
    f.set(a[v]);
}

void main() {
    read(n), read(m), read(q), sqn = sqrt(n);
    for (int i = 2, x; i <= n; i++) read(x), G[x].push_back(i);
    for (int i = 1; i <= n; i++) read(a[i]);
    dfs(1), dfs(1, 1);
    for (int i = 1, t, ans; i <= q; i++) {
        read(t), read(c[1]), lca = c[1];
        for (int i = 2; i <= t; i++) read(c[i]), lca = get_lca(lca, c[i]);
        for (int i = 1; i <= t; i++) get_bitset(c[i], lca, f[i]);
        ans = m / t + 1;
        for (int x = 1; x < (1 << t) && ans; x++) {
            mix.reset();
            for (int i = 1; i <= t; i++) if (x & (1 << (i - 1))) mix |= f[i];
            ans = std::min(ans, (int)mix.count() / __builtin_popcount(x));
        }
        print(ans * t, '\n');
    }
}

```



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**

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

[分块](#)[树链剖分](#)[Hall 定理](#)[bitset](#)

可以在这里写评论哦 ~

使用 **Hexo Hey** 进行 Hexo 下的写作
[上一篇](#) «

BZOJ5405 platform
» [下一篇](#)



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

© 2017 - 2019 [memset0](#) 的博客.

[浙ICP备19006255号-1](#)

97691 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](#)

