树相关问题

2017211123 褚逸豪

2018 年 4 月 10 日

1　满k叉树相关问题

一棵满k叉树按层次顺序从1开始对全部结点编号，若所求结点存在，则有以下问题

1.1　各层结点的数目为多少

第i层有k^{i-1}个结点

1.2　编号为n的结点的父结点的编号是多少

设编号为n的结点的父结点的编号为$f(n)$，则有

$$f(n) = \left\lfloor \frac{n-2}{k} \right\rfloor + 1$$

1.3　结点n的第i个儿子的编号是多少

设编号为n的结点的第i个儿子的编号为$g(n,i)$，则有

$$g(n,i) = (n-1)k + i + 1$$

1.4　结点n有右兄弟的条件是什么

显然需要满足以下条件

$$n \not\equiv 1 \pmod k$$

2　度为m的树的相关问题

已知一棵度为m的树中有n_i个度为i的结点，则有以下问题

2.1　该树有多少个叶子节点

显然答案为n_0

04/10/18 04:56:13 D:\git-repos\data-structure-homework\06\e15.cpp

```cpp
#include <cstdio>
#include <cstring>
#include <iostream>
#define MXN 1007
using namespace std;

struct node {
    int dep, id;
    node *fa[11];
    node *c[2];
    node() { dep = id = 0; }
} nds[MXN];
void dfs(node *root, node *f) {
    if (!root || !(root - nds)) return;
    root->dep = f->dep + 1;
    root->fa[0] = f;
    dfs(root->c[0], root);
    dfs(root->c[1], root);
}
node *lca(node *x, node *y) {
    if (x->dep > y->dep) swap(x, y);
    for (int d = y->dep - x->dep, i = 0; d; ++i, d >>= 1)
        if (d & 1)
            y = y->fa[i];
    if (x == y) return x;
    for (int i = 10; i >= 0; --i)
        if (x->fa[i] != y->fa[i])
            x = x->fa[i], y = y->fa[i];
    return x->fa[0];
}
int n;
int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        int a, b;
        nds[i].id = i;
        scanf("%d %d", &a, &b);
        nds[i].c[0] = nds + a;
        nds[i].c[1] = nds + b;
    }
    nds[0].fa[0] = nds;
    dfs(nds + 1, nds);
    for (int j = 1; j < 11; ++j) {
        for (int i = 0; i <= n; ++i) {
            nds[i].fa[j] = nds[i].fa[j - 1]->fa[j - 1];
        }
    }
    int a, b;
    while (~scanf("%d %d", &a, &b)) {
        printf("The LCA is %d\n", lca(nds + a, nds + b)->id);
    }
}
/**
d:\git-repos\data-structure-homework\06>g++ e15.cpp

d:\git-repos\data-structure-homework\06>a
7 2 3 4 5 6 7 0 0 0 0 0 0 0 0
1 2
The LCA is 1
1 3
The LCA is 1
4 5
The LCA is 2
4 6
The LCA is 1
5 7
The LCA is 1
2 6
The LCA is 1
^Z

d:\git-repos\data-structure-homework\06>
*/
```

04/10/18 04:56:25 D:\git-repos\data-structure-homework\06\e16.cpp

```
#include <cstdio>
#include <cstring>
#include <iostream>
#define MXN 1007
using namespace std;
int seq[MXN];
struct node {
    int dep, id;
    node *fa[11];
    node *c[2];
    node() { dep = id = 0; }
} nds[MXN];
int mxdep = 0;
void dfs(node *root, node *f) {
    if (!root || !(root - nds)) return;
    root->dep = f->dep + 1;
    mxdep = max(mxdep, root->dep);
    root->fa[0] = f;
    dfs(root->c[0], root);
    dfs(root->c[1], root);
}
bool ddfs(node *root) {
    if (mxdep == root->dep) {
        seq[root->dep] = root->id;
        return true;
    }
    if (ddfs(root->c[0])) {
        seq[root->dep] = root->id;
        return true;
    } else if (ddfs(root->c[1])) {
        seq[root->dep] = root->id;
        return true;
    }
    return false;
}
int n;
int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        int a, b;
        nds[i].id = i;
        scanf("%d %d", &a, &b);
        nds[i].c[0] = nds + a;
        nds[i].c[1] = nds + b;
    }
    nds[0].fa[0] = nds;
    dfs(nds + 1, nds);
    ddfs(nds + 1);
    printf("The height of this tree is %d, following is a sequence of nodes\n", mxdep);
    for (int i = 1; i <= mxdep; ++i)
        printf("%d ", seq[i]);
    putchar('\n');
}
/**
d:\git-repos\data-structure-homework\06>g++ e16.cpp

d:\git-repos\data-structure-homework\06>a
7 2 3 4 5 6 7 0 0 0 0 0 0 0 0
The height of this tree is 3, following is a sequence of nodes
1 2 4

d:\git-repos\data-structure-homework\06>
*/
```

04/10/18 04:56:49 D:\git-repos\data-structure-homework\06\e17.cpp

```cpp
#include <cstdio>
#include <cstring>
#include <iostream>
#define MXN 1007
using namespace std;
struct node {
    int id;
    node *c[2];
    node() { id = 0; }
} nds[MXN];
void dfs(node *root) {
    if (!root || !(root - nds)) return;
    putchar('(');
    dfs(root->c[0]);
    printf("%d", root->id);
    dfs(root->c[1]);
    putchar(')');
}

int n;
int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        int a, b;
        nds[i].id = i;
        scanf("%d %d", &a, &b);
        nds[i].c[0] = nds + a;
        nds[i].c[1] = nds + b;
    }
    dfs(nds + 1);
}
/**
d:\git-repos\data-structure-homework\06>g++ e17.cpp

d:\git-repos\data-structure-homework\06>a
7 2 3 4 5 6 7 0 0 0 0 0 0 0 0
(((4)2(5))1((6)3(7)))
d:\git-repos\data-structure-homework\06>
*/
```