

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

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

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

```

1  #include <stdio>
2  #include <cstring>
3  #include <iostream>
4  #define MXN 1007
5  using namespace std;
6
7  struct node {
8      int dep, id;
9      node *fa[11];
10     node *c[2];
11     node() { dep = id = 0; }
12 } nds[MXN];
13 void dfs(node *root, node *f) {
14     if (!root || !(root - nds)) return;
15     root->dep = f->dep + 1;
16     root->fa[0] = f;
17     dfs(root->c[0], root);
18     dfs(root->c[1], root);
19 }
20 node *lca(node *x, node *y) {
21     if (x->dep > y->dep) swap(x, y);
22     for (int d = y->dep - x->dep, i = 0; d; ++i, d >>= 1)
23         if (d & 1)
24             y = y->fa[i];
25     if (x == y) return x;
26     for (int i = 10; i >= 0; --i)
27         if (x->fa[i] != y->fa[i])
28             x = x->fa[i], y = y->fa[i];
29     return x->fa[0];
30 }
31 int n;
32 int main() {
33     scanf("%d", &n);
34     for (int i = 1; i <= n; ++i) {
35         int a, b;
36         nds[i].id = i;
37         scanf("%d %d", &a, &b);
38         nds[i].c[0] = nds + a;
39         nds[i].c[1] = nds + b;
40     }
41     nds[0].fa[0] = nds;
42     dfs(nds + 1, nds);
43     for (int j = 1; j < 11; ++j) {
44         for (int i = 0; i <= n; ++i) {
45             nds[i].fa[j] = nds[i].fa[j - 1]->fa[j - 1];
46         }
47     }
48     int a, b;
49     while (~scanf("%d %d", &a, &b)) {
50         printf("The LCA is %d\n", lca(nds + a, nds + b)->id);
51     }
52 }
53 /**
54 d:\git-repos\data-structure-homework\06>g++ e15.cpp
55
56 d:\git-repos\data-structure-homework\06>a
57 7 2 3 4 5 6 7 0 0 0 0 0 0 0 0
58 1 2
59 The LCA is 1
60 1 3
61 The LCA is 1
62 4 5
63 The LCA is 2
64 4 6
65 The LCA is 1
66 5 7
67 The LCA is 1
68 2 6
69 The LCA is 1
70 ^Z
71
72 d:\git-repos\data-structure-homework\06>
73 */

```

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

```

1  #include <stdio>
2  #include <cstring>
3  #include <iostream>
4  #define MXN 1007
5  using namespace std;
6  int seq[MXN];
7  struct node {
8      int dep, id;
9      node *fa[11];
10     node *c[2];
11     node() { dep = id = 0; }
12 } nds[MXN];
13 int mxdep = 0;
14 void dfs(node *root, node *f) {
15     if (!root || !(root - nds)) return;
16     root->dep = f->dep + 1;
17     mxdep = max(mxdep, root->dep);
18     root->fa[0] = f;
19     dfs(root->c[0], root);
20     dfs(root->c[1], root);
21 }
22 bool ddfs(node *root) {
23     if (mxdep == root->dep) {
24         seq[root->dep] = root->id;
25         return true;
26     }
27     if (ddfs(root->c[0])) {
28         seq[root->dep] = root->id;
29         return true;
30     } else if (ddfs(root->c[1])) {
31         seq[root->dep] = root->id;
32         return true;
33     }
34     return false;
35 }
36 int n;
37 int main() {
38     scanf("%d", &n);
39     for (int i = 1; i <= n; ++i) {
40         int a, b;
41         nds[i].id = i;
42         scanf("%d %d", &a, &b);
43         nds[i].c[0] = nds + a;
44         nds[i].c[1] = nds + b;
45     }
46     nds[0].fa[0] = nds;
47     dfs(nds + 1, nds);
48     ddfs(nds + 1);
49     printf("The height of this tree is %d, following is a sequence of nodes\n", mxdep);
50     for (int i = 1; i <= mxdep; ++i)
51         printf("%d ", seq[i]);
52     putchar('\n');
53 }
54 /**
55 d:\git-repos\data-structure-homework\06>g++ e16.cpp
56
57 d:\git-repos\data-structure-homework\06>a
58 7 2 3 4 5 6 7 0 0 0 0 0 0 0
59 The height of this tree is 3, following is a sequence of nodes
60 1 2 4
61
62 d:\git-repos\data-structure-homework\06>
63 */

```

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

```

1  #include <cstdio>
2  #include <cstring>
3  #include <iostream>
4  #define MXN 1007
5  using namespace std;
6  struct node {
7      int id;
8      node *c[2];
9      node() { id = 0; }
10 } nds[MXN];
11 void dfs(node *root) {
12     if (!root || !(root - nds)) return;
13     putchar('(');
14     dfs(root->c[0]);
15     printf("%d", root->id);
16     dfs(root->c[1]);
17     putchar(')');
18 }
19
20 int n;
21 int main() {
22     scanf("%d", &n);
23     for (int i = 1; i <= n; ++i) {
24         int a, b;
25         nds[i].id = i;
26         scanf("%d %d", &a, &b);
27         nds[i].c[0] = nds + a;
28         nds[i].c[1] = nds + b;
29     }
30     dfs(nds + 1);
31 }
32 /**
33 d:\git-repos\data-structure-homework\06>g++ e17.cpp
34
35 d:\git-repos\data-structure-homework\06>a
36 7 2 3 4 5 6 7 0 0 0 0 0 0 0
37 (((4)2(5))1((6)3(7)))
38 d:\git-repos\data-structure-homework\06>
39 */

```