LCA

人员

阮文璋、褚锦轩、王承周、许睿谦 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/2240

比赛概况 | 题目列表 | 选择题列表 | 提交记录 | ★ 实时榜单 | ★ 选择题排行榜

王向东老师周日一点半C++综合练习

刷新

#	用户名	姓名	编程分	时间	A	B	C
1	ruanwenzhang	阮文璋	300	3075	100	100	100
2	wangchengzhou	王承周	300	9425	100	100	100
3	chenxinmiao	陈欣妙	271	2268	100	100	71
4	xuruiqian	许睿谦	106	1443	100	6	
5	chujinxuan	褚锦轩	100	166	100		
6	wangyibo	王毅博	100	1404	100		

本周作业

https://cppoj.kids123code.com/contest/2331 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天讲了 LCA 这个知识点, 同学们课上整体理解的都比较好。

课堂内容

Good Tuple Problem (上周作业)

dfs 黑白染色判定二分图

```cpp
#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
vector<int> vec[maxn];
int a[maxn], b[maxn], col[maxn];
bool flag = true;

void dfs(int u, int value) {
  if (!flag) return;
  if (col[u] == value) return;
  if (col[u]!=-1 && col[u]!=value) flag = false;
```

```
  col[u] = value;
  for (int i : vec[u]) dfs(i, 1-value);
}

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= m; ++i) cin >> a[i];
  for (int i = 1; i <= m; ++i) {
    cin >> b[i];
    vec[a[i]].push_back(b[i]), vec[b[i]].push_back(a[i]);
  }

  memset(col, -1, sizeof(col));
  for (int i = 1; i <= n; ++i) {
    if (col[i] != -1) continue;
    dfs(i, 0);
    if (!flag) { cout << "No" << endl; return 0; }
  }
  cout << "Yes" << endl;
  return 0;
}
```

【模板】最近公共祖先（LCA）

倍增法求 最近公共祖先

f[i][j]: i 这个点, 往上 2^j 的父亲是谁

(a,b) 的最近公共祖先: 先倍增往上跳, 直到两个点变到一层; 然后倍增往上判, 只要两个点不重合就往上, 最后返回 2 个点的父结点

```
#include <bits/stdc++.h>

using namespace std;

const int N = 5e5 + 5, M = 20;
vector<int> vec[N];
int dis[N], f[N][M];

void dfs(int u, int fa, int d) {
  dis[u] = d; f[u][0] = fa;
  for (int i = 1; i < M; ++i) f[u][i] = f[f[u][i-1]][i-1];
  for (int i : vec[u]) {
    if (i == fa) continue;
    dfs(i, u, d+1);
  }
}

int lca(int a, int b) {
  if (dis[a] < dis[b]) return lca(b, a);
```

```
  for (int i = M-1; i >= 0; --i) {
    if (dis[f[a][i]] >= dis[b]) a = f[a][i];
  }

  if (a == b) return a;

  for (int i = M-1; i >= 0; --i) {
    if (f[a][i] != f[b][i]) a = f[a][i], b = f[b][i];
  }
  return f[a][0];
}

int main()
{
  ios::sync_with_stdio(false);
  cin.tie(0);

  int n, m, root; cin >> n >> m >> root;
  for (int i = 1; i <= n-1; ++i) {
    int a, b; cin >> a >> b;
    vec[a].push_back(b), vec[b].push_back(a);
  }

  dfs(root, 0, 1);

  while (m -- ) {
    int a, b; cin >> a >> b;
    cout << lca(a, b) << "\n";
  }
  return 0;
}
```

[GESP202312 八级] 大量的工作沟通

dfs 中, 维护 ans[i]: 从根到 i 中最大的编号是多少

找到所有点的 LCA 点 id, 答案即为 ans[id]

```
#include <bits/stdc++.h>

using namespace std;

const int N = 1e5 + 5, M = 20;
vector<int> vec[N];
int dis[N], f[N][M], ans[N];

void dfs(int u, int fa, int d) {
  dis[u] = d; f[u][0] = fa; ans[u] = max(ans[fa], u);
  for (int i = 1; i < M; ++i) f[u][i] = f[f[u][i-1]][i-1];
  for (int i : vec[u]) {
```

```
    if (i == fa) continue;
    dfs(i, u, d+1);
  }
}

int lca(int a, int b) {
  if (dis[a] < dis[b]) return lca(b, a);

  for (int i = M-1; i >= 0; --i) {
    if (dis[f[a][i]] >= dis[b]) a = f[a][i];
  }

  if (a == b) return a;

  for (int i = M-1; i >= 0; --i) {
    if (f[a][i] != f[b][i]) a = f[a][i], b = f[b][i];
  }
  return f[a][0];
}

void solve() {
  int m; cin >> m;
  int fa; cin >> fa; ++fa;
  for (int i = 2; i <= m; ++i) {
    int x; cin >> x; ++x; fa = lca(fa, x);
  }

  cout << ans[fa] - 1 << endl;
}

int main()
{
  int n; cin >> n;
  for (int i = 2; i <= n; ++i) {
    int id; cin >> id; ++id; vec[id].push_back(i);
  }

  dfs(1, 0, 1);

  int T; cin >> T;
  while (T -- ) solve();
  return 0;
}
```

[蓝桥杯 2025 国 Python A] 巡逻

维护 2 个数组

dis[i]: 从根到 i, 中间的点数

p[i]: 从根到 i, 中间的风险和

树上任意两点 i j 之间的距离为: dis[i] + dis[j] - 2*dis[fa]

树上任意两点 i j 之间的路径和为: p[i] + p[j] - 2*p[fa]

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

const int N = 5000 + 5, M = 20;
struct node {
  int to, value;
};
vector<node> vec[N];
int dis[N], f[N][M], p[N];

void dfs(int u, int fa, int d) {
  dis[u] = d; f[u][0] = fa;
  for (int i = 1; i < M; ++i) f[u][i] = f[f[u][i-1]][i-1];
  for (node it : vec[u]) {
    if (it.to == fa) continue;
    p[it.to] = p[u] + it.value;
    dfs(it.to, u, d+1);
  }
}

int lca(int a, int b) {
  if (dis[a] < dis[b]) return lca(b, a);

  for (int i = M-1; i >= 0; --i) {
    if (dis[f[a][i]] >= dis[b]) a = f[a][i];
  }

  if (a == b) return a;

  for (int i = M-1; i >= 0; --i) {
    if (f[a][i] != f[b][i]) a = f[a][i], b = f[b][i];
  }
  return f[a][0];
}

signed main()
{
  int n, K; cin >> n >> K;
  for (int i = 1; i <= n-1; ++i) {
    int a, b, c; cin >> a >> b >> c;
    vec[a].push_back({b,c}), vec[b].push_back({a,c});
  }

  dfs(1, 0, 1);

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= n; ++j) {
      int fa = lca(i, j);
```

```
        if (dis[i]+dis[j]-2*dis[fa] == K) res += p[i]+p[j]-2*p[fa];
      }
    }
    cout << res << endl;
    return 0;
}
```