```cpp
#include "../bits/stdc++.h"
// グラフ，根を引数として初期化
// 蟻本準拠
// varified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3215160
class Lca
{
  int n, log_n;
  std::vector<std::vector<int>> parent;
  std::vector<int> depth;
  void dfs(const std::vector<std::vector<int>> &g, int v, int p, int d)
  {
    parent[0][v] = p;
    depth[v] = d;
    for (int i = 0; i < (int)g[v].size(); i++)
    {
      if (g[v][i] != p)
        dfs(g, g[v][i], v, d + 1);
    }
  }

public:
  Lca(const std::vector<std::vector<int>> &g, int root) : n((int)g.size()), log_n(0), depth(n, 0)
  {
    for (int v = n; v > 0; v /= 2)
      log_n++;
    parent.assign(log_n, std::vector<int>(n, 0));
    dfs(g, root, -1, 0);
    for (int k = 0; k < log_n - 1; k++)
    {
      for (int v = 0; v < n; v++)
      {
        if (parent[k][v] < 0)
          parent[k + 1][v] = -1;
        else
          parent[k + 1][v] = parent[k][parent[k][v]];
      }
    }
  }
  int query(int u, int v)
  {
    if (depth[u] > depth[v])
      std::swap(u, v);
    for (int k = 0; k < log_n; k++)
    {
      if (((depth[v] - depth[u]) >> k) & 1)
      {
        v = parent[k][v];
      }
    }
    if (u == v)
      return u;
    for (int k = log_n - 1; k >= 0; k--)
    {
      if (parent[k][u] != parent[k][v])
      {
        u = parent[k][u];
        v = parent[k][v];
      }
    }
    return parent[0][u];
  }
};

```