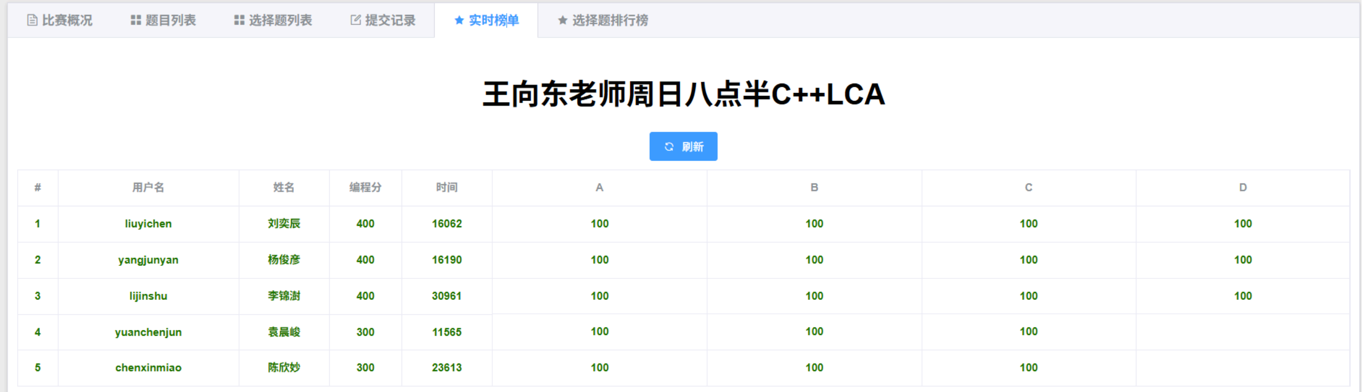
树上差分+01bfs

人员

刘奕辰、杨俊彦、袁晨峻、李锦澍、陈欣妙 到课

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/2329



王向东老师周日八点半C++LCA

#	用户名	姓名	编程分	时间	A	B	C	D
1	liuyichen	刘奕辰	400	16062	100	100	100	100
2	yangjunyan	杨俊彦	400	16190	100	100	100	100
3	lijinshu	李锦澍	400	30961	100	100	100	100
4	yuanchenjun	袁晨峻	300	11565	100	100	100	
5	chenxinmiao	陈欣妙	300	23613	100	100	100	

本周作业

https://cppoj.kids123code.com/contest/2453 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天讲了 树上差分 和 01bfs 这两个小知识点, 同学们课上听课吸收整体都不错。

课堂内容

[蓝桥杯 2022 国 B] 机房 (上周作业)

对每个点的 deg 维护一个前缀和, 然后后续用 LCA 就可以 O(1) 求了

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

using namespace std;

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

void dfs(int u, int fa, int d) {
  dis[u] = d; f[u][0] = fa; p[u] = p[fa] + deg[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];
}

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

  dfs(1, 0, 1);

  while (m -- ) {
    int a, b; cin >> a >> b;
    if (a == b) cout << deg[a] << endl;
    else {
      int fa = lca(a, b);
      cout << p[a] + p[b] - 2*p[fa] + deg[fa] << endl;
    }
  }
  return 0;
}
```

[USACO15DEC] Max Flow P

树上差分, 对树上的 u~v 路径进行 +1 时, 可以找到它们的最近公共祖先 id 点

对 w[u]++, w[v]++, w[id]--, w[fa[id][0]]-- 即可

最后做一遍 dfs, 把子节点的 w 信息自底向上维护一下

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

using namespace std;

```

```
const int N = 5e4 + 5, M = 20;
vector<int> vec[N];
int dis[N], f[N][M], w[N];

void dfs(int u, int fa) {
  dis[u] = dis[fa]+1, 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) dfs(i, u);
  }
}

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 dfs2(int u, int fa) {
  for (int i : vec[u]) {
    if (i == fa) continue;
    dfs2(i, u);
    w[u] += w[i];
  }
}

int main()
{
  int n, m; cin >> n >> m;
  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(1, 0);

  while (m -- ) {
    int a, b; cin >> a >> b;
    int u = lca(a, b), fu = f[u][0];
    ++w[a], ++w[b], --w[u], --w[fu];
  }

  dfs2(1, 0);
  int res = 0;
  for (int i = 1; i <= n; ++i) res = max(res, w[i]);
  cout << res << endl;
  return 0;
}
```

[USACO12MAR] Tractor S

01bfs, 用 deque 维护即可, 边权为 0 的往前放, 边权为 1 的往后放

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

using namespace std;

const int maxn = 1000 + 5;
struct node {
  int x, y;
};
int dis[maxn][maxn], c[maxn][maxn];
bool st[maxn][maxn];
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};

int main()
{
  int n, x0, y0; cin >> n >> x0 >> y0;
  while (n -- ) {
    int x, y; cin >> x >> y; ++c[x][y];
  }

  memset(dis, 0x3f, sizeof(dis));
  deque<node> dq; dq.push_back({x0,y0}); dis[x0][y0] = 0;
  while (!dq.empty()) {
    node u = dq.front(); dq.pop_front();
    int x = u.x, y = u.y;
    if (st[x][y]) continue;
    st[x][y] = true;

    for (int i = 0; i < 4; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=0 && nx<maxn && ny>=0 && ny<maxn && dis[x][y]+c[nx][ny]<dis[nx][ny])
{
        dis[nx][ny] = dis[x][y]+c[nx][ny];
        if (!c[nx][ny]) dq.push_front({nx,ny});
        else dq.push_back({nx,ny});
      }
    }
  }

  cout << dis[0][0] << endl;
  return 0;
}
```

[GESP202503 八级] 割裂

树上差分, 把好点对之间的路径都进行 +1 操作, 把坏点对之间的路径都进行 -1 操作, 最后看有哪些点是 -1 即可

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

using namespace std;

const int N = 1e6 + 5, M = 20;
vector<int> vec[N];
int dis[N], f[N][M], w[N];

void dfs(int u, int fa) {
  dis[u] = dis[fa]+1, 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) dfs(i, u);
  }
}

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 dfs2(int u, int fa) {
  for (int i : vec[u]) {
    if (i == fa) continue;
    dfs2(i, u);
    w[u] += w[i];
  }
}

int main()
{
  int n, m; cin >> n >> m;
  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(1, 0);

  for (int i = 1; i <= m+1; ++i) {
    int a, b; cin >> a >> b;
    int u = lca(a, b), fu = f[u][0];
    if (i <= m) ++w[a], ++w[b], --w[u], --w[fu];
    else --w[a], --w[b], ++w[u], ++w[fu];
  }

  dfs2(1, 0);

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    if (w[i] < 0) ++res;
  }
  cout << res << endl;
  return 0;
}
```