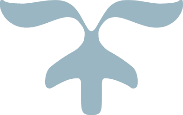


DAA WEEK – 14 SKILL – 14



# [The Story of a Tree](https://www.hackerrank.com/contests/daa-skill-14-graphs-part-ii/challenges/the-story-of-a-tree)

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#define MAX 100007

int edge[MAX][100];

int edge\_count[MAX];

int csum[MAX], st[MAX], ed[MAX], parent[MAX];

int indx;

void add\_edge(int u, int v) {

edge[u][edge\_count[u]++] = v;

edge[v][edge\_count[v]++] = u;

}

void dfs(int idx, int par) {

parent[idx] = par;

indx++;

st[idx] = indx;

for (int i = 0; i < edge\_count[idx]; i++) {

int x = edge[idx][i];

if (x == par) continue;

dfs(x, idx);

}

ed[idx] = indx;

}

int gcd(int a, int b) {

return b == 0 ? a : gcd(b, a % b);

}

int main() {

int t, n, u, v, q, k;

scanf("%d", &t);

while (t--) {

memset(csum, 0, sizeof(csum));

memset(edge\_count, 0, sizeof(edge\_count));

indx = 0;

int valid = 0;

scanf("%d", &n);

for (int i = 1; i < n; i++) {

scanf("%d %d", &u, &v);

add\_edge(u, v);

}

int root = 1;

dfs(root, -1);

scanf("%d %d", &q, &k);

for (int i = 1; i <= q; i++) {

scanf("%d %d", &u, &v);

if (parent[v] == u) {

csum[st[root]]++;

csum[ed[root] + 1]--;

csum[st[v]]--;

csum[ed[v] + 1]++;

} else {

csum[st[u]]++;

csum[ed[u] + 1]--;

}

}

for (int i = st[root]; i <= ed[root]; i++) {

csum[i] += csum[i - 1];

if (csum[i] >= k)

valid++;

}

int g = gcd(valid, n);

printf("%d/%d\n", valid / g, n / g);

}

return 0;

}

**The Story of a Tree Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Prim's (MST) : Special Subtree](https://www.hackerrank.com/contests/daa-skill-14-graphs-part-ii/challenges/primsmstsub)

#include <stdio.h>

#include <stdlib.h>

struct E {

int u, v, w;

};

struct S {

int p, r;

};

int find(struct S s[], int i) {

if (s[i].p != i) {

s[i].p = find(s, s[i].p);

}

return s[i].p;

}

void uni(struct S s[], int x, int y) {

int xr = find(s, x);

int yr = find(s, y);

if (s[xr].r < s[yr].r) {

s[xr].p = yr;

} else if (s[xr].r > s[yr].r) {

s[yr].p = xr;

} else {

s[yr].p = xr;

s[xr].r++;

}

}

int cmp(const void\* a, const void\* b) {

return ((struct E\*)a)->w - ((struct E\*)b)->w;

}

int main() {

int n, m;

scanf("%d %d", &n, &m);

struct E\* e = malloc(m \* sizeof(struct E));

for (int i = 0; i < m; i++) {

scanf("%d %d %d", &e[i].u, &e[i].v, &e[i].w);

}

qsort(e, m, sizeof(e[0]), cmp);

struct S\* s = malloc(n \* sizeof(struct S));

for (int i = 0; i < n; i++) {

s[i].p = i;

s[i].r = 0;

}

int tw = 0, count = 0;

for (int i = 0; i < m && count < n - 1; i++) {

int x = find(s, e[i].u - 1);

int y = find(s, e[i].v - 1);

if (x != y) {

tw += e[i].w;

uni(s, x, y);

count++;

}

}

printf("%d\n", tw);

free(e);

free(s);

return 0;

}

**Prim's (MST) : Special Subtree Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

**SKILL WEEK – 14**

<https://www.hackerrank.com/contests/daa-skill-14-graphs-part-ii/challenges>