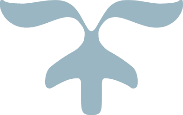


DAA WEEK – 14 SKILL – 14



# [Crab Graphs](https://www.hackerrank.com/contests/daa-skill-15-graphs-part-iii/challenges/crab-graphs)

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

struct edge {

int a, b, c;

} es[42000];

#define MAX 0x3fffffff

int estop;

int adj[210][420];

int adj\_size[210];

int cur[210];

int dis[210];

int n, m;

int s, t;

void addedge(int u, int v, int c) {

adj[u][adj\_size[u]++] = estop;

es[estop++] = (struct edge){u, v, c};

adj[v][adj\_size[v]++] = estop;

es[estop++] = (struct edge){v, u, 0};

}

int BFS() {

int q[210];

memset(dis, -1, sizeof(dis));

int be = 0, ed = 1;

q[0] = s;

dis[s] = 0;

while (be < ed) {

int l = adj\_size[q[be]];

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

struct edge e = es[adj[q[be]][i]];

if (e.c > 0 && dis[e.b] == -1) {

dis[e.b] = dis[q[be]] + 1;

q[ed++] = e.b;

}

}

be++;

}

return (dis[t] != -1);

}

int dinic(int x, int maxflow) {

if (x == t)

return maxflow;

int l = adj\_size[x];

for (int i = cur[x]; i < l; i++) {

cur[x] = i;

struct edge \*e = &es[adj[x][i]];

if (dis[e->b] == dis[x] + 1 && e->c > 0) {

int f = dinic(e->b, (maxflow < e->c ? maxflow : e->c));

if (f) {

e->c -= f;

es[adj[x][i] ^ 1].c += f;

return f;

}

}

}

return 0;

}

int dinic\_main() {

int F = 0;

while (BFS()) {

memset(cur, 0, sizeof(cur));

int f;

while ((f = dinic(s, MAX)))

F += f;

}

return F;

}

int main() {

int tt;

scanf("%d", &tt);

while (tt--) {

int T;

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

estop = 0;

s = 0;

t = 2 \* n + 1;

for (int i = 0; i <= 2 \* n + 1; i++) {

adj\_size[i] = 0;

}

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

int a, b;

scanf("%d %d", &a, &b);

addedge(2 \* a - 1, 2 \* b, 1);

addedge(2 \* b - 1, 2 \* a, 1);

}

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

addedge(0, 2 \* i + 1, T);

addedge(2 \* i + 2, 2 \* n + 1, 1);

}

printf("%d\n", dinic\_main());

}

return 0;

}

**Crab Graphs Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Jeanie's Route](https://www.hackerrank.com/contests/daa-skill-15-graphs-part-iii/challenges/jeanies-route)

#include <stdio.h>

#include <stdlib.h>

#define Maxn 100001

int n, k, x, y, z, G[Maxn], good[Maxn], pr[Maxn], ans, pas, D, goodsub[Maxn];

struct Node {

int vertex;

int weight;

struct Node\* next;

};

struct Node\* g[Maxn];

struct Node\* createNode(int v, int w) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

newNode->vertex = v;

newNode->weight = w;

newNode->next = NULL;

return newNode;

}

void addEdge(int u, int v, int w) {

struct Node\* node = createNode(v, w);

node->next = g[u];

g[u] = node;

}

void dfs(int v, int p) {

struct Node\* temp = g[v];

while (temp != NULL) {

if (temp->vertex != p) {

dfs(temp->vertex, v);

if (goodsub[temp->vertex]) goodsub[v] = 1;

}

temp = temp->next;

}

if (good[v]) goodsub[v] = 1;

}

int dfs\_for\_diameter(int v, int p) {

int max1 = 0, max2 = 0;

struct Node\* temp = g[v];

while (temp != NULL) {

if (temp->vertex != p && goodsub[temp->vertex]) {

ans += 2 \* temp->weight;

int d = dfs\_for\_diameter(temp->vertex, v) + temp->weight;

if (d > max1) {

max2 = max1;

max1 = d;

} else if (d > max2) {

max2 = d;

}

}

temp = temp->next;

}

if (max1 + max2 > D) D = max1 + max2;

return max1;

}

int main() {

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

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

scanf("%d", &G[i]);

good[G[i]] = 1;

}

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

scanf("%d %d %d", &x, &y, &z);

addEdge(x, y, z);

addEdge(y, x, z);

}

dfs(G[1], 0);

dfs\_for\_diameter(G[1], 0);

printf("%d", ans - D);

return 0;

}

**Jeanie's Route Test CasesA screenshot of a computer

AI-generated content may be incorrect.**

**SKILL WEEK – 15**

<https://www.hackerrank.com/contests/daa-skill-15-graphs-part-iii/challenges>