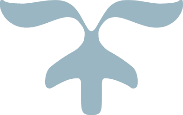


DAA WEEK – 13 SKILL – 13



# [Breadth First Search: Shortest Reach](https://www.hackerrank.com/contests/daa-skill-13-graphs-part-i/challenges/bfsshortreach/problem)

# ( JAVA 7 )

import java.io.\*;

import java.util.\*;

public class Solution {

public static ArrayList<Integer> bfs(ArrayList<ArrayList<Integer>> adj, int s) {

LinkedList<Integer> q = new LinkedList<Integer>();

ArrayList<Integer> result = new ArrayList<Integer>(adj.size());

for (int i = 0; i < adj.size(); i++) {

result.add(0);

}

q.addFirst(s);

while (q.size() > 0) {

int current = q.pollLast();

ArrayList<Integer> tmp = adj.get(current);

for (int i = 0; i < tmp.size(); i++) {

int v = tmp.get(i);

if (result.get(v) == 0) {

q.addFirst(v);

result.set(v, result.get(current) + 6);

}

}

}

return result;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int q = in.nextInt();

for (int a0 = 0; a0 < q; a0++) {

int n = in.nextInt();

ArrayList<ArrayList<Integer>> adj = new ArrayList<ArrayList<Integer>>(n);

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

adj.add(new ArrayList<Integer>());

}

int m = in.nextInt();

for (int a1 = 0; a1 < m; a1++) {

int u = in.nextInt() - 1;

int v = in.nextInt() - 1;

ArrayList<Integer> tmp = adj.get(u);

tmp.add(v);

adj.set(u, tmp);

tmp = adj.get(v);

tmp.add(u);

adj.set(v, tmp);

}

int s = in.nextInt() - 1;

ArrayList<Integer> result = Solution.bfs(adj, s);

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

if (i != s) {

if (result.get(i) == 0) {

System.out.print("-1 ");

} else {

System.out.print(result.get(i) + " ");

}

}

}

System.out.print("\n");

}

in.close();

}

}

**Breadth First Search: Shortest Reach Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Kruskal (MST): Really Special Subtree](https://www.hackerrank.com/contests/daa-skill-13-graphs-part-i/challenges/kruskalmstrsub)

#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;

}

**Kruskal (MST): Really Special Subtree Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

**SKILL WEEK – 13**

<https://www.hackerrank.com/daa-skill-13-graphs-part-i>