**Obilazak grafa pomoću BFS u grafu sa vektorima susedstva, C++**

vector <int> cv[N+1];

queue <int> q;

int posecen[N+1];

**Breadth First Search** algoritam posećuje sve susede jednog čvora (one koji su udaljeni jednu vezu od njega) zatim sve njihove prve susede (tj. One koji su udaljeni dve veze od početnog čvora) i.t.d. dogod ima neposećenih suseda. Algoritam tako ide u **širinu** dokle god može a najlakše se implementira pomoću reda (kontejner queue).

Za svakog suseda čvora koji je na početku reda, poseti sve njegove susede koji do sada nisu posećeni, označi ih da su posećeni i stavi ih u red.

void bfs(int x){

memset(posecen,0,sizeof(posecen)); // isprazni niz posecenosti

posecen[x]=true; // označi čvor x da je posećen

q.push(x); // stavi čvor x u red

while (q.size()) { // dogod ima članova u redu

x=q.front(); // obradjuemo onog ko je prvi u redu

for (int i=0; i<v[x].size(); i++) // za svakog suseda cvora x (i-ti sused)

if (!posecen[v[x][i]]) { // ako nije posećen

q.push(v[x][i]); // stavi ga na kraj reda

posecen[v[x][i]]=true; // i označi ga da je posećen

}

q.pop(); // x je odradjen, skidam ga s pocetka reda.

}

}

Bfs se može pustiti iz bilo kog čvora i on će posetiti sve čvorove do kojih se iz tog čvora može stići. Recimo

bfs(1);

Ako graf nije povezan, tada se koristi FullDfs. To je puštanje dfs algoritma iz svih čvorova koji nisu posećeni iz prethodnih dfs-a.  
  
for (int i=1; i<=N; i++) // iz svakog čvora grafa

if (!posecen(i)) // koji nije posećen do sada

bfs(i); // pusti bfs

Koristan je kada treba da se izračunaju udaljenosti u grafu.