Grafos #1: Flood Fill

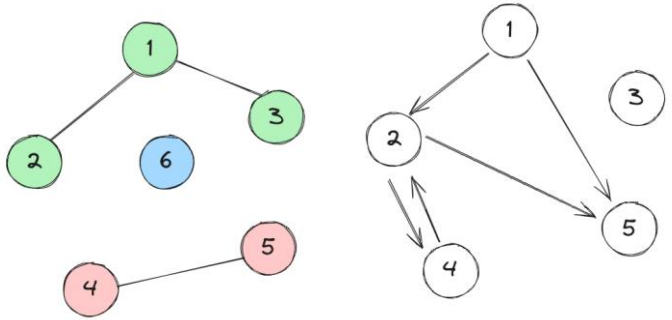
Mateo Álvarez Murillo
Universidad Nacional de Colombia, Sede Medellín
Abril 26, 2024

Contenido

- Terminología
- Representación en código
- BFS
- DFS
- Flood Fill
- Problemas vistos
- Recursos

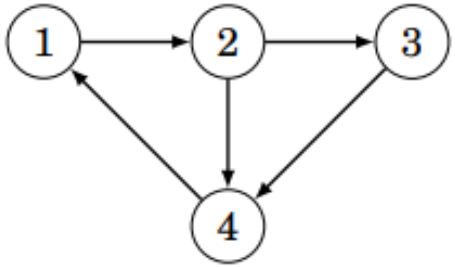
Terminología

- Grafo: Estructura matemática que representa un conjunto de objetos llamados vértices (o nodos), conectados por enlaces llamados aristas.
- Vértice (Nodo): Punto individual en un grafo que puede tener conexiones con otros vértices.
- Arista: Conexión entre dos vértices en un grafo. Puede ser dirigida o no dirigida, dependiendo de si la conexión tiene una dirección específica o no.



Representación en código

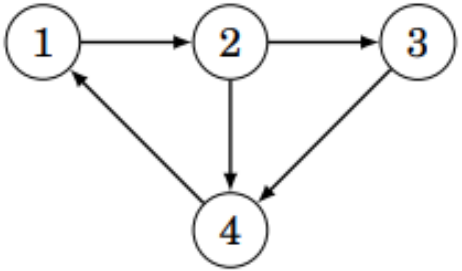
```cpp
// forma 1
vector<vector<int>> matrix1 = {
    {0, 1, 0, 0},
    {0, 0, 1, 1},
    {0, 0, 0, 1},
    {1, 0, 0, 0}
};

// forma 2
vector<vector<int>> matrix2(n, vector<int>(n,0));
cin>>a>>b;
matrix2[a][b]=true;
```



	1	2	3	4
1	0	1	0	0
2	0	0	1	1
3	0	0	0	1
4	1	0	0	0

Representación en código

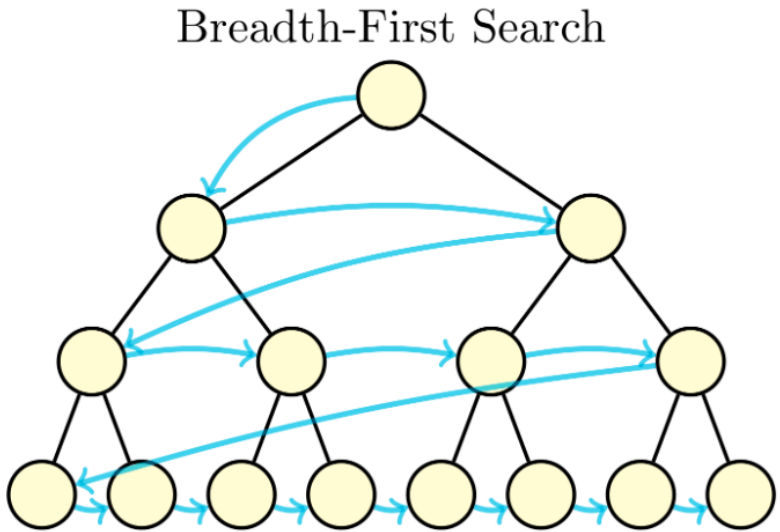
```cpp
// forma 1
vector<vector<int>> lista1 = {
    {2},    // nodo 1
    {3, 4}, // nodo 2
    {4},    // nodo 3
    {1}     // nodo 4
};

// forma 2
vector<vector<int>> lista2 (n, vector<int>());
cin>>a>>b;
lista2[a].push_back(b);
```



1 : {2}
2: {3, 4}
3: {4}
4: {1}

BFS (Breadth-First Search) O(n+m)

Algoritmo de búsqueda utilizado para explorar o recorrer un grafo o una estructura similar desde un nodo inicial, visitando todos los nodos vecinos antes de pasar a los siguientes niveles de vecindad.



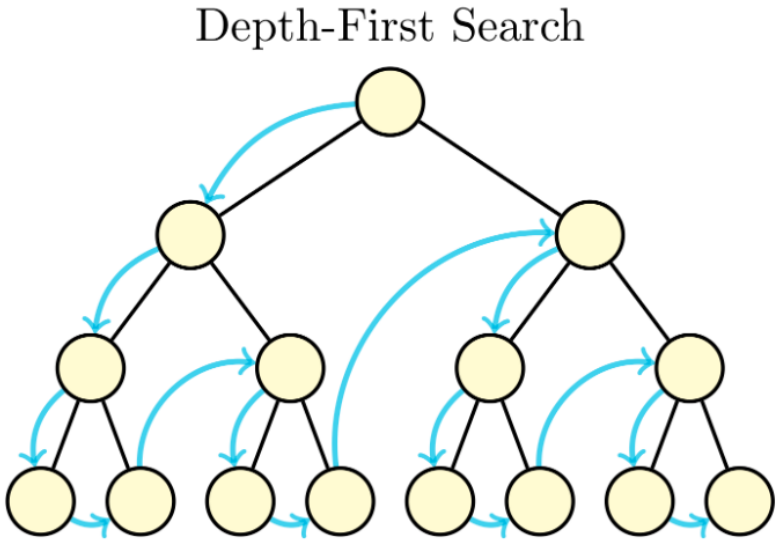
BFS (Breadth-First Search)

```python
n = int(input())
adj = [[] for _ in range(n)]
visited = [False for _ in range(n)]

def bfs(x):
    queue = deque([x])
    while queue:
        s = queue.popleft()
        # process node s
        for u in adj[s]:
            if visited[u]:
                continue
            visited[u] = True
            queue.append(u)
```

```cpp
vector<vector<int>> adj;
vector<bool> visited;
int n;

void bfs(int x){
    queue<int> q;
    q.push(x);
    visited[x] = true;
    while (!q.empty()) {
        int s = q.front(); q.pop();
        // process node s
        for (auto u : adj[s]) {
            if (visited[u]) continue;
            visited[u] = true;
            q.push(u);
        }
    }
}
```

DFS (Depth-First Search) O(n+m)

Algoritmo de búsqueda utilizado para explorar o recorrer un grafo o una estructura similar desde un nodo inicial, yendo tan lejos como sea posible por un camino antes de retroceder y continuar explorando otros caminos.



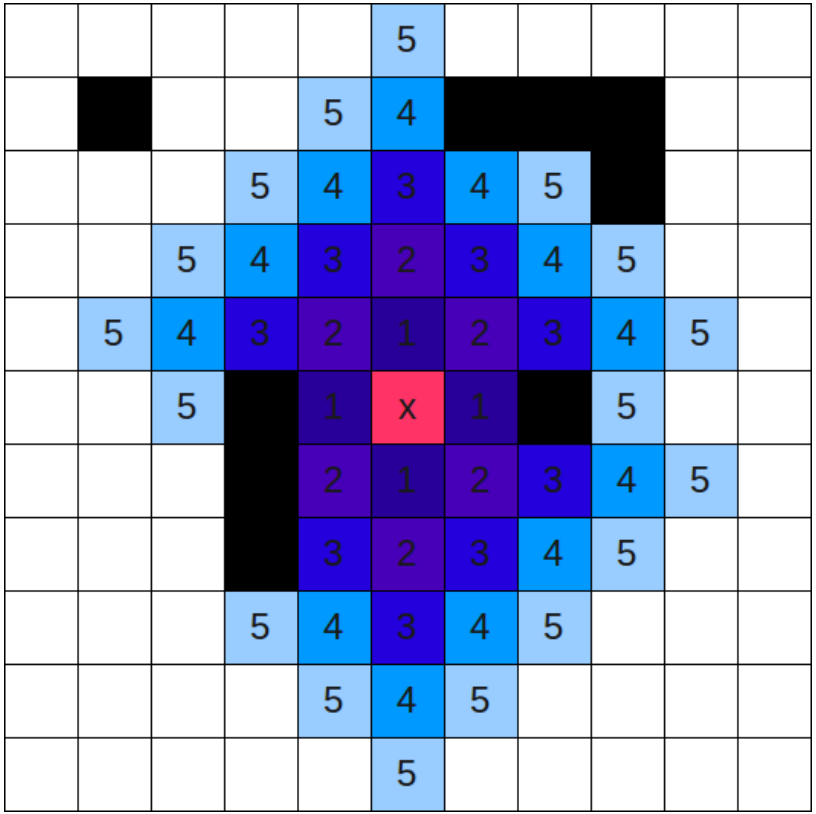
DFS (Depth-First Search)

```python
n = int(input())
adj = [[] for _ in range(n)]
visited = [False for _ in range(n)]

def dfs(v):
    visited[v] = True
    # process node v
    for u in adj[v]:
        if not visited[u]:
            dfs(u)
```

```cpp
vector<vector<int>> adj;
vector<bool> visited;
int n;

void dfs(int v){
    visited[v]=true;
    // process node v
    for(auto u:adj[v]){
        if(!visited[u]){
            dfs(u);
        }
    }
}
```

Flood Fill
o(r*c)

Algoritmo utilizado para colorear una matriz o navegar dentro de ella como si fuera un grafo. Comienza desde un punto de origen y se propaga a través de los vecinos desde la casilla inicial.



Flood Fill

```cpp
const int dirx[4] = {0,1,0,-1};
const int diry[4] = {1,0,-1,0};
vector<vector<bool>> visited;
vector<vector<int>> mapa;
int n,m;

void bfs(int x1, int y1){
    queue<pair<int, int>> q;
    q.push({x1, y1});
    visited[x1][y1] = true;
    while (!q.empty()) {
        auto act = q.front(); q.pop();
        x1 = act.first;
        y1 = act.second;
        for (int i=0;i<4;++i) {
            int x2 = x1 + dirx[i];
            int y2 = y1 + diry[i];
            if (x2 < 0 || x2 ≥ n || y2 < 0 || y2 ≥ m){
                continue;
            }
            if (visited[x2][y2]){
                continue;
            }
            visited[x2][y2]=true;
            q.push({x2,y2});
        }
    }
}
```

Problemas vistos

- https://cses.fi/problemset/task/1666
- https://cses.fi/problemset/task/1668/
- https://cses.fi/problemset/task/1192
- https://cses.fi/problemset/task/1194

Recursos

- https://usaco.guide/silver/graph-traversal
- https://usaco.guide/silver/flood-fill
- https://cses.fi/book/index.php
- https://www.youtube.com/watch?v=VuiXOc81UDM&t=327s&ab_channel=Insidecode
- https://www.youtube.com/watch?v=JWP9EI88Yoo&ab_channel=Insidecode
- https://www.youtube.com/watch?v=gohEAP1Jmcg&ab_channel=Insidecode