

# Grafos

# Travessias ## Descrição - dfs: (Depth-First Search) Travessia por profundidade complexidade  $O(N)$  sendo N o número de arestas do vértice u - bfs: (Breadth-First Search) Travessia por largura; Armazena no vetor dist, a distância do vertice u para os demais vertice; o próprio vértice e vértices que não foram visitados tem distância 0; complexidade  $O(N)$  sendo N o número de arestas do vértice u

```
#define MAXN 1009
bitset<MAXN> visited;
vector<int> adj_list[MAXN];

void dfs(int u){
    visited[u]=true;
    for(auto v:adj_list[u]){
        if(not visited[v]){
            dfs(v);
        }
    }
}

int dist[MAXN];
void bfs(int u){
    queue<int> q;
    q.push(u);
    visited[u]=true;
    dist[u]=0;
    while(not q.empty()){
        auto x = q.front();
        q.pop();
        for(auto v:adj_list[x]){
            if(visited[v]) continue;

            visited[v]=true;
            dist[v] = dist[x]+1;
            q.push(v);
        }
    }
}
```