

# Grafos

*Componentes Conectados*

**Prof. Edson Alves**

***Faculdade UnB Gama***

## **Grafos conectados**

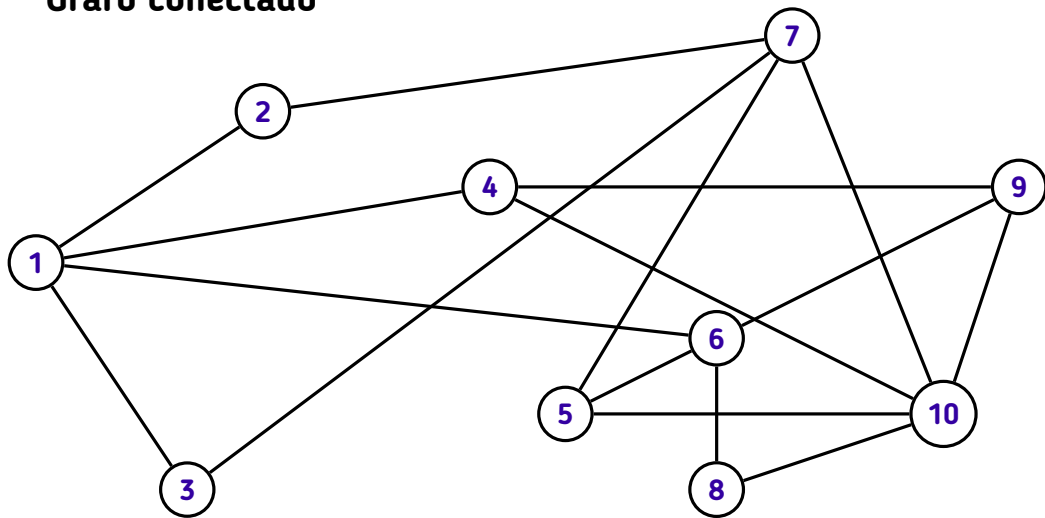
## Grafos conectados

Um grafo não-direcionado  $G(V, E)$  é dito **conectado** se, para qualquer

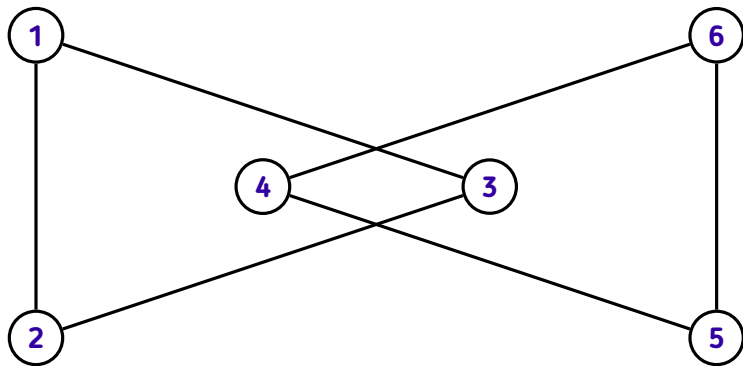
## Grafos conectados

Um grafo não-direcionado  $G(V, E)$  é dito **conectado** se, para qualquer par de vértices  $u, v \in V$ , existe ao menos um caminho de  $u$  a  $v$ .

## Grafo conectado



## Grafo não-conectado



**Componente conectado**

## Componente conectado

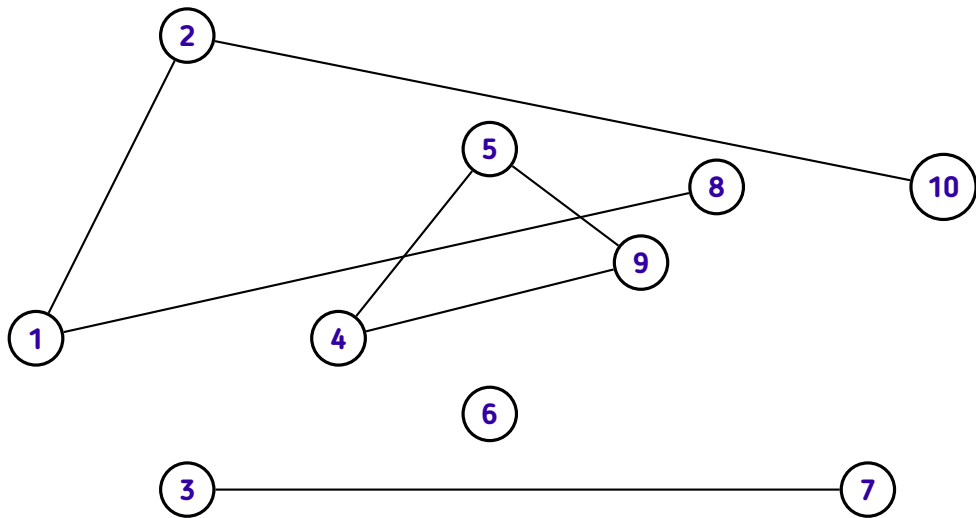
O componente conectado do grafo  $G(V, E)$  que contém o vértice  $u$  é o maior subgrafo conectado  $S(V', E')$  de  $G$  tal que  $u \in V'$

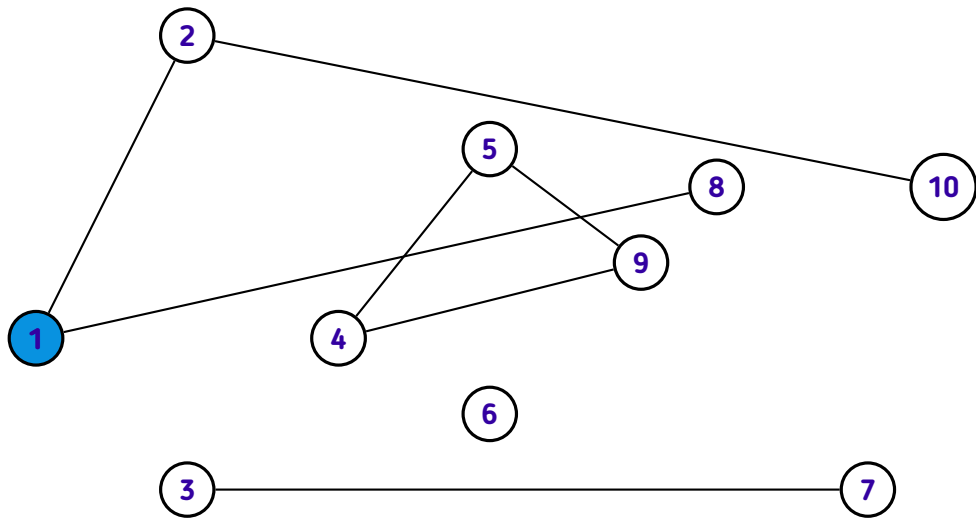


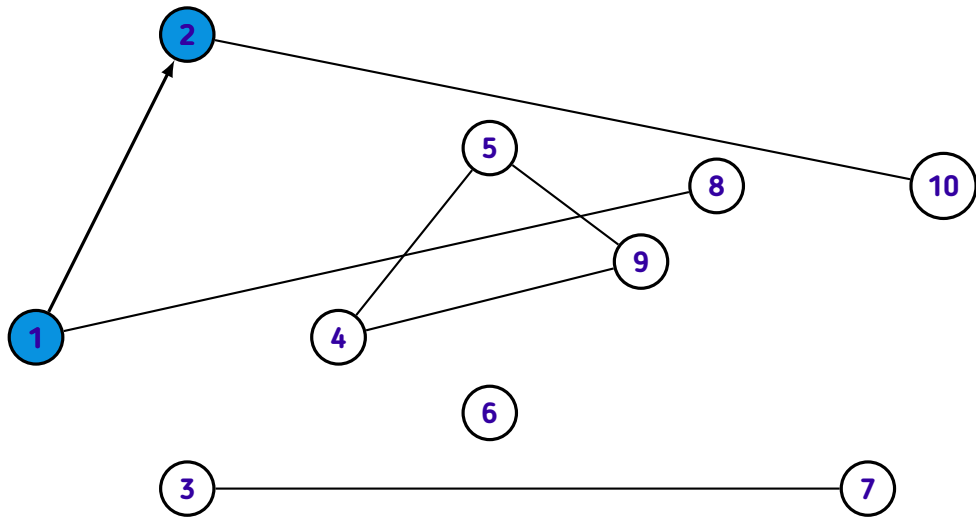
## Componente conectado

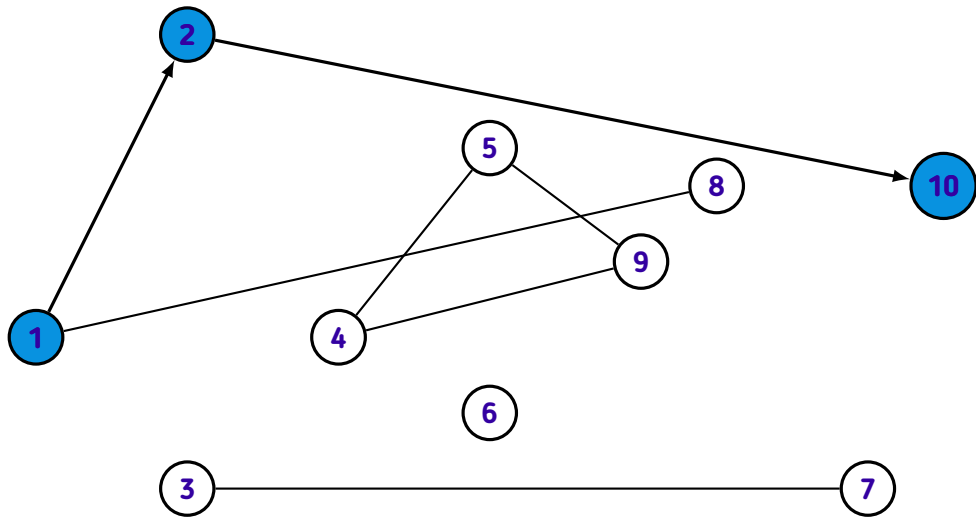
O componente conectado do grafo  $G(V, E)$  que contém o vértice  $u$  é o maior subgrafo conectado  $S(V', E')$  de  $G$  tal que  $v \in V'$

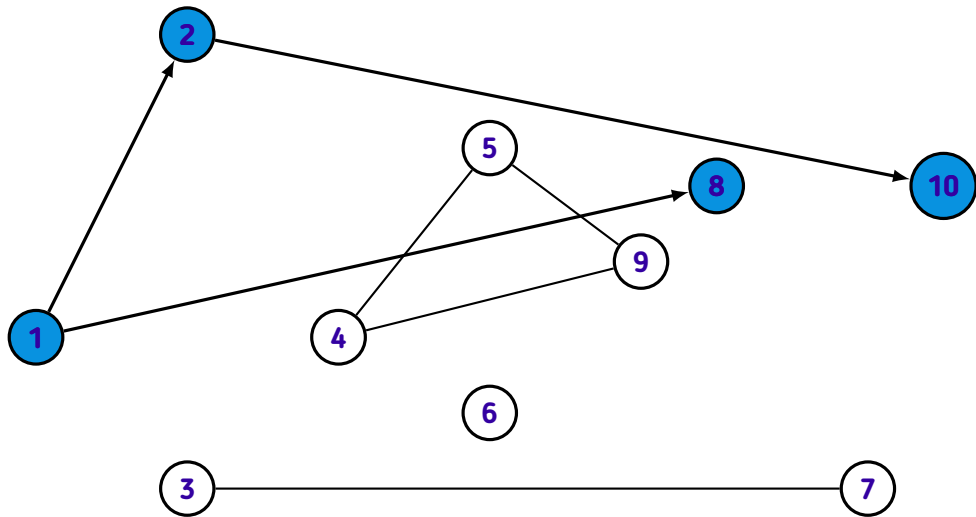
Os elementos de  $V'$  podem ser determinados por meio de uma travessia com início em  $v$ .

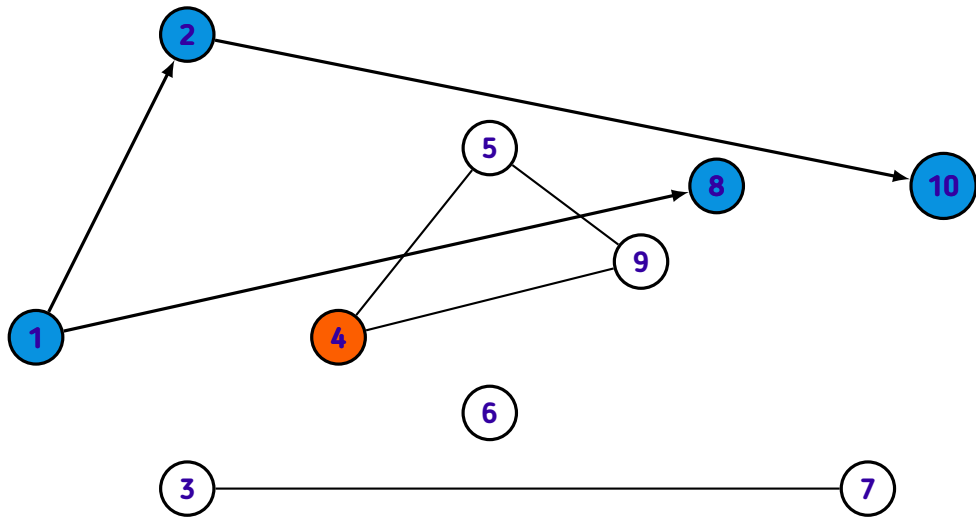


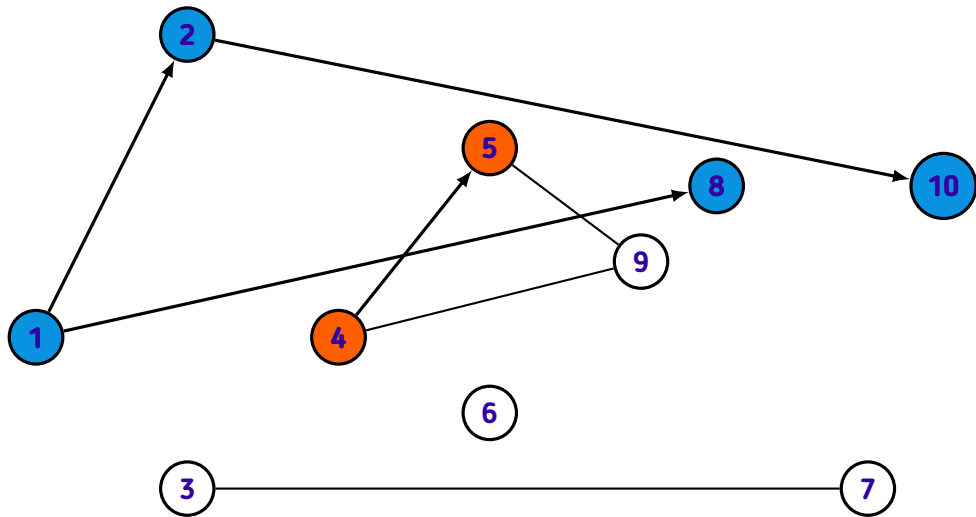




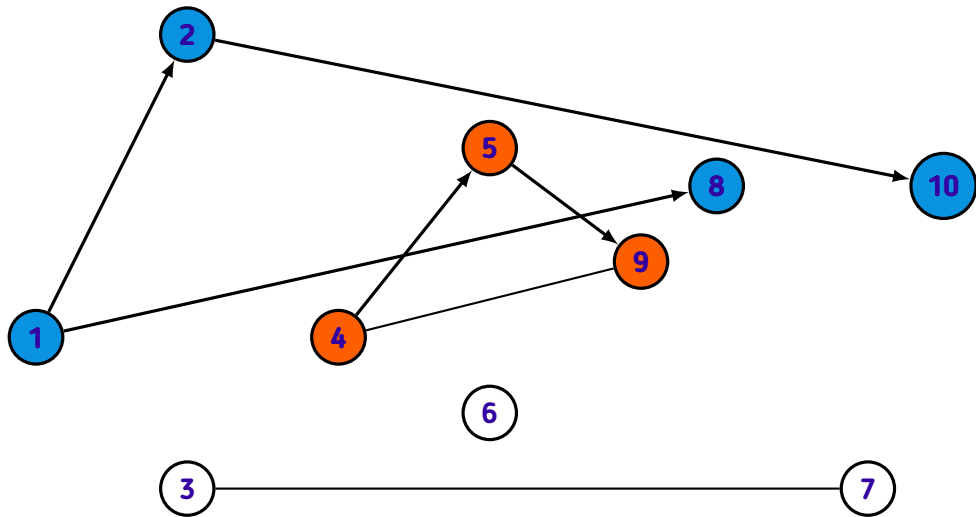


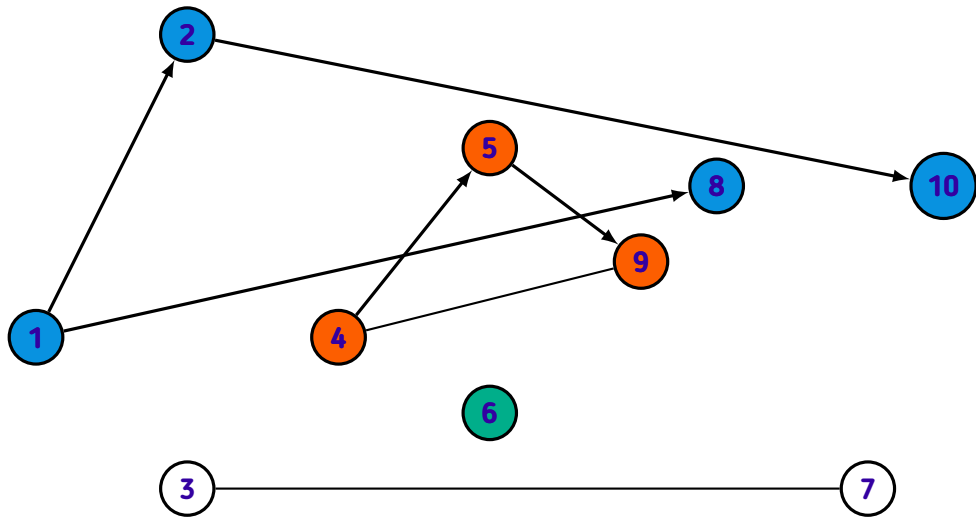


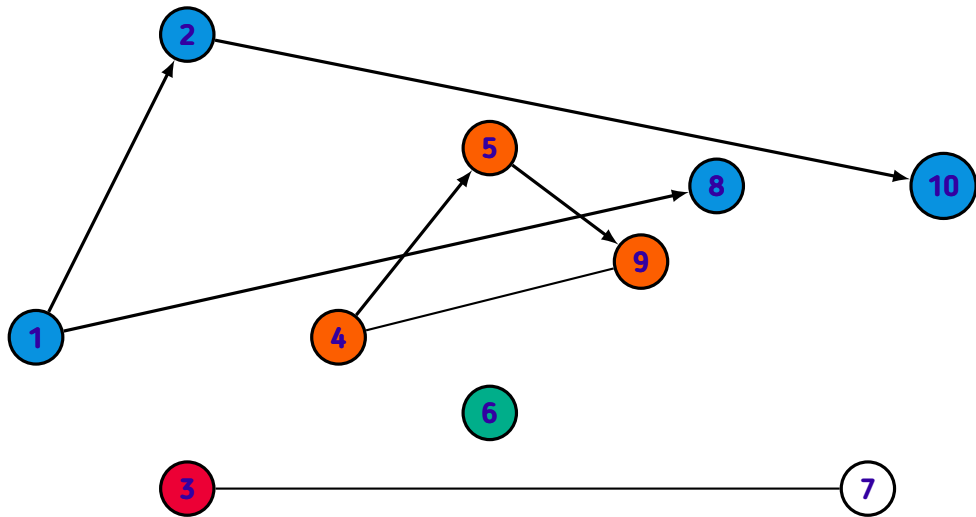


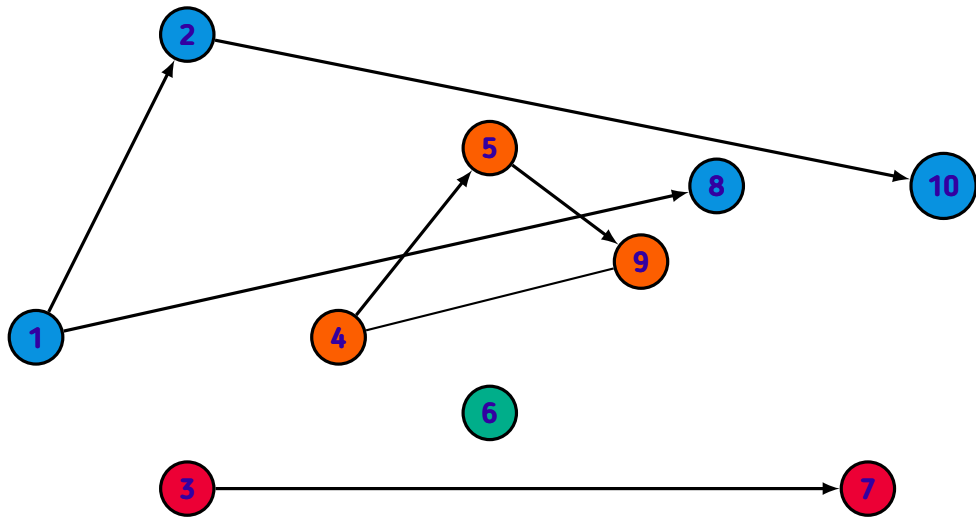












```
int connected_components(int N)
{
    visited.reset();

    int count = 0;

    for (int u = 1; u <= N; ++u)
    {
        if (not visited[u])
        {
            cout << "Component " << ++count << ":";
            dfs(u);
            cout << '\n';
        }
    }

    return count;
}
```

```
void dfs(int u)
{
    if (visited[u])
        return;

    visited[u] = true;

    cout << ' ' << u;

    for (auto v : adj[u])
        dfs(v);
}
```

## **Grafos conectados e componentes conectados**

## Grafos conectados e componentes conectados

Um grafo não-direcionado  $G$  é conectado se, e somente se,  $G$  tem um único componente conectado.



## Problemas sugeridos

1. [AtCoder Beginner Contest 049 – Problem D: Connectivity](#)
2. [Educational Codeforces Round 5 – Problem C: The Labyrinth](#)
3. [Educational Codeforces Round 33 \(Rated for Div. 2\) – Problem C: Rumor](#)
4. [OJ 11094 – Continents](#)

## Referências

1. HALIM, Felix; HALIM, Steve. *Competitive Programming 3*, 2010.
2. LAAKSONEN, Antti. *Competitive Programmer's Handbook*, 2018.
3. SKIENA, Steven; REVILLA, Miguel. *Programming Challenges*, 2003.





