Grafos

Representação de grafos

Prof. Edson Alves

Faculdade UnB Gama

Matriz de adjacências

Matriz de adjacências

- $\star$ **Seja G um grafo com N vértices**

Matriz de adjacências

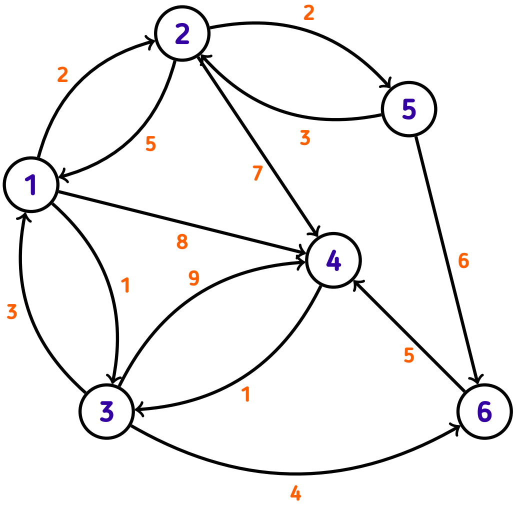
- ⋆ Seja G um grafo com N vértices
- ⋆ Assuma que cada vértice seja associado a um inteiro positivo em $[1, N]$

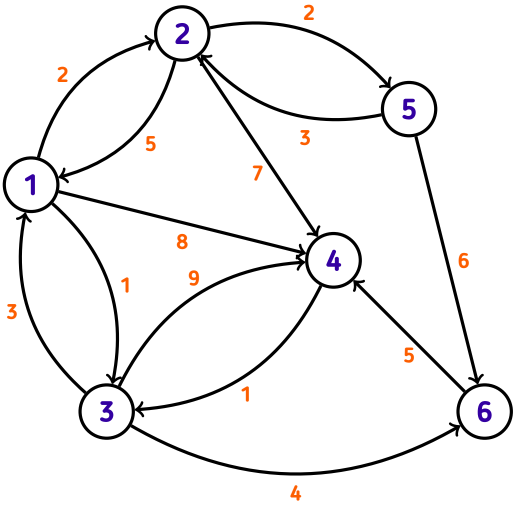
Matriz de adjacências

- $\star$ **Seja G um grafo com N vértices**
- $\star$ **Assuma que cada vértice seja associado a um inteiro positivo em $[1, N]$**
- $\star$ **Na matrix de adjacências $A_{N \times N}$ o elemento a_{ij} é o peso da aresta (i, j)**

Matriz de adjacências

- $\star$ Seja G um grafo com N vértices
- $\star$ Assuma que cada vértice seja associado a um inteiro positivo em $[1, N]$
- $\star$ Na matrix de adjacências $A_{N \times N}$ o elemento a_{ij} é o peso da aresta (i, j)
- $\star$ Se $(i, j) \notin E$, então $a_{ij} = 0$


2
2
5
1
5
3
7
8
4
9
1
6
3
5
1
3
6
4



$$A = \begin{bmatrix} 0 & 2 & 1 & 8 & 0 & 0 \\ 5 & 0 & 0 & 7 & 2 & 0 \\ 3 & 0 & 0 & 9 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 5 & 0 & 0 \end{bmatrix}$$

Características das matrizes de adjacências

Características das matrizes de adjacências

- $\star$ **Se G é não poderado,** $a_{ij} \in [0, 1]$

Características das matrizes de adjacências

- $\star$ **Se G é não poderado, $a_{ij} \in [0, 1]$**
- $\star$ **Se G é multigrafo, a_{ij} pode registrar o número de ocorrências de (i, j)**

Características das matrizes de adjacências

- $\star$ **Se** G **é não poderado,** $a_{ij} \in [0, 1]$
- $\star$ **Se** G **é multigrafo,** a_{ij} **pode registrar o número de ocorrências de** (i, j)
- $\star$ **Se** G **é simples,** $a_{ii} = 0, \forall i \in V$

Características das matrizes de adjacências

- ⋆ **Se G é não poderado, $a_{ij} \in [0, 1]$**
- ⋆ **Se G é multigrafo, a_{ij} pode registrar o número de ocorrências de (i, j)**
- ⋆ **Se G é simples, $a_{ii} = 0, \forall i \in V$**
- ⋆ **Vantagem: Consulta "$(i, j) \in E$?" respondida em $O(1)$**

Características das matrizes de adjacências

- $\star$ **Se G é não poderado, $a_{ij} \in [0, 1]$**
- $\star$ **Se G é multigrafo, a_{ij} pode registrar o número de ocorrências de (i, j)**
- $\star$ **Se G é simples, $a_{ii} = 0, \forall i \in V$**
- $\star$ **Vantagem: Consulta "$(i, j) \in E$?" respondida em $O(1)$**
- $\star$ **Desvantagem: Complexidade de memória $O(N^2)$**

```cpp
#include <bits/stdc++.h>

const int N { 6 };
int A[N + 1][N + 1];

int main()
{
    A[1][2] = 2, A[1][3] = 1, A[1][4] = 8;
    A[2][1] = 5, A[2][4] = 7, A[2][5] = 2;
    A[3][1] = 3, A[3][4] = 9, A[3][6] = 4;
    A[4][3] = 1;
    A[5][2] = 3, A[5][6] = 6;
    A[6][4] = 5;

    for (int i = 1; i <= N; ++i)
        for (int j = 1; j <= N; ++j)
            std::cout << A[i][j] << (j == N ? '\n' : ' ');

    return 0;
}
```

Lista de adjacências

Lista de adjacências

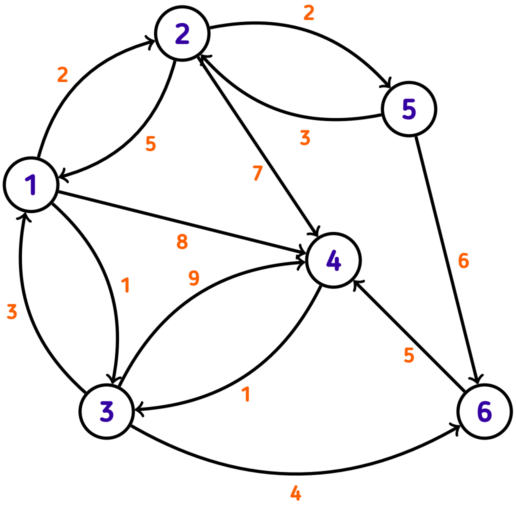
- A cada vértice u é associada uma lista $l(u)$

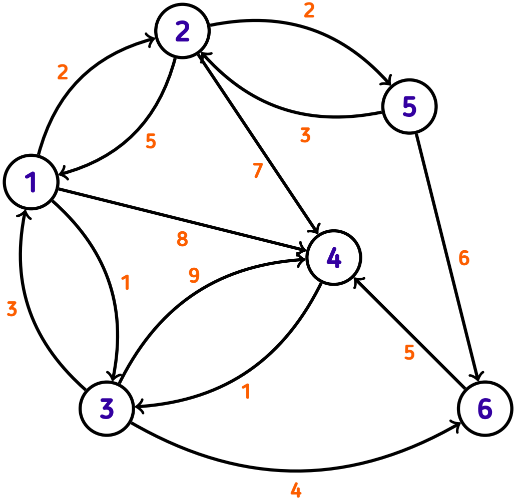
Lista de adjacências

- $\star$ **A cada vértice** u **é associada uma lista** $l(u)$
- $\star$ **Esta lista contém os vértices** v **tais que** $(u, v) \in E$

Lista de adjacências

- $\star$ **A cada vértice u é associada uma lista $l(u)$**
- $\star$ **Esta lista contém os vértices v tais que $(u, v) \in E$**
- $\star$ **Se G é ponderado, então os elementos de $l(u)$ são pares (v_i, w_i)**


2
2
5
5
3
7
1
8
6
9
1
3
4
5
1
3
6
4



1	(2, 2) (3, 1) (4, 8)
2	(1, 5) (4, 7) (5, 2)
3	(1, 3) (4, 9) (6, 4)
4	(3, 1)
5	(2, 3) (6, 6)
6	(4, 5)

```cpp
#include <bits/stdc++.h>

using namespace std;
using ii = pair<int, int>;

vector<ii> adj[] { {}, { { 2, 2 }, { 3, 1 }, { 4, 8 } },
    { { 1, 5 }, { 4, 7 }, { 5, 2 } }, { { 1, 3 }, { 4, 9 }, { 6, 4 } },
    { { 3, 1 } }, { { 2, 3 }, { 6, 6 } }, { { 4, 5 } }, };

int main()
{
    for (int u = 1; u <= 6; ++u) {
        cout << u << ":";

        for (auto [v, w] : adj[u])
            cout << " (" << v << ", " << w << ')';
        cout << '\n';
    }

    return 0;
}
```

Características das listas de adjacências

Características das listas de adjacências

- **Possíveis listas em C++:** `list`, `vector` **ou** `forward_list`

Características das listas de adjacências

- **Possíveis listas em C++:** `list`, `vector` **ou** `forward_list`
- **A escolha depende de como as arestas serão acessadas**

Características das listas de adjacências

- ⋆ **Possíveis listas em C++:** `list`, `vector` **ou** `forward_list`
- ⋆ **A escolha depende de como as arestas serão acessadas**
- ⋆ **Complexidade de memória:** $O(N + M)$, **onde** M **é o número de arestas**

Características das listas de adjacências

- ⋆ **Possíveis listas em C++:** `list`, `vector` **ou** `forward_list`
- ⋆ **A escolha depende de como as arestas serão acessadas**
- ⋆ **Complexidade de memória:** $O(N + M)$, **onde** M **é o número de arestas**
- ⋆ **São adequadas para grafos esparsos**

Características das listas de adjacências

- ⋆ **Possíveis listas em C++:** `list`, `vector` **ou** `forward_list`
- ⋆ **A escolha depende de como as arestas serão acessadas**
- ⋆ **Complexidade de memória:** $O(N + M)$, **onde** M **é o número de arestas**
- ⋆ **São adequadas para grafos esparsos**
- ⋆ **Algoritmos clássicos utilizam esta representação**

Lista de arestas

Lista de arestas

- $\star$ O grafo G é representado pelo conjunto de arestas E

Lista de arestas

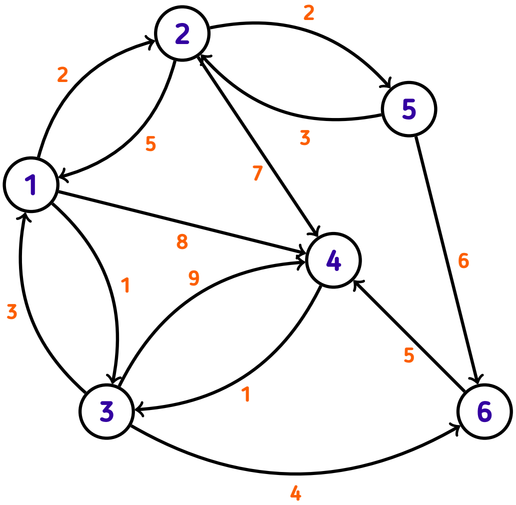
- ⋆ **O grafo G é representado pelo conjunto de arestas E**
- ⋆ **Cada aresta é representada pela tripla (u, v, w)**

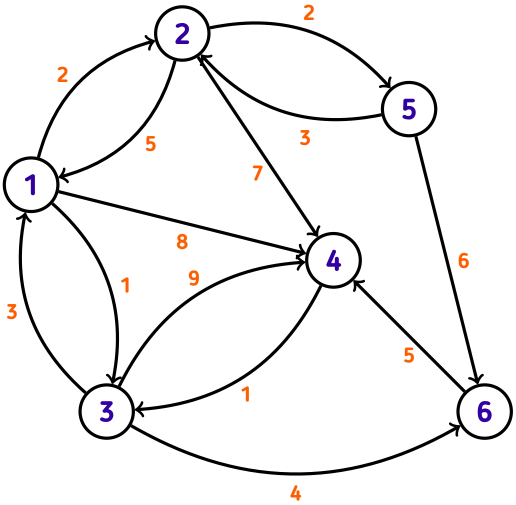
Lista de arestas

- ⋆ O grafo G é representado pelo conjunto de arestas E
- ⋆ Cada aresta é representada pela tripla (u, v, w)
- ⋆ É possível deduzir V a partir de E se G não tem vértices isolados

Lista de arestas

- ⋆ O grafo G é representado pelo conjunto de arestas E
- ⋆ Cada aresta é representada pela tripla (u, v, w)
- ⋆ É possível deduzir V a partir de E se G não tem vértices isolados
- ⋆ Complexidade de memória: $O(M)$


2
2
5
3
7
1
8
9
1
3
6
5
1
4
2
5
1
4
3
6



(1, 2, 2)
(1, 3, 1)
(1, 4, 8)
(2, 1, 5)
(2, 4, 7)
(2, 5, 2)
(3, 1, 3)
(3, 4, 9)
(3, 6, 4)
(4, 3, 1)
(5, 2, 3)
(5, 6, 6)
(6, 4, 5)

```cpp
#include <bits/stdc++.h>

using namespace std;

using edge = tuple<int, int, int>;

vector<edge> es { { 1, 2, 2 }, { 1, 3, 1 }, { 1, 4, 8 }, { 2, 1, 5 },
    { 2, 4, 7 }, { 2, 5, 2 }, { 3, 1, 3 }, { 3, 4, 9 }, { 3, 6, 4 },
    { 4, 3, 1 }, { 5, 2, 3 }, { 5, 6, 6 }, { 6, 4, 5 } };

int main()
{
    for (const auto& [u, v, w] : es)
        cout << "(" << u << ", " << v << ", " << w << ")\n";

    return 0;
}
```

Representação implícita

Representação implícita

- Os vértices e as arestas são definidas por relações

Representação implícita

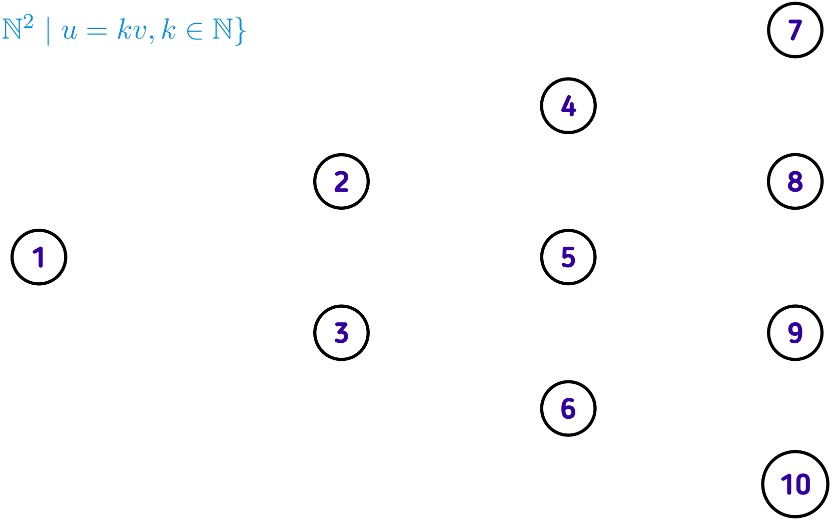
- Os vértices e as arestas são definidas por relações
- Adequada para grafos complexos ou com infinitos vértices e arestas

Representação implícita

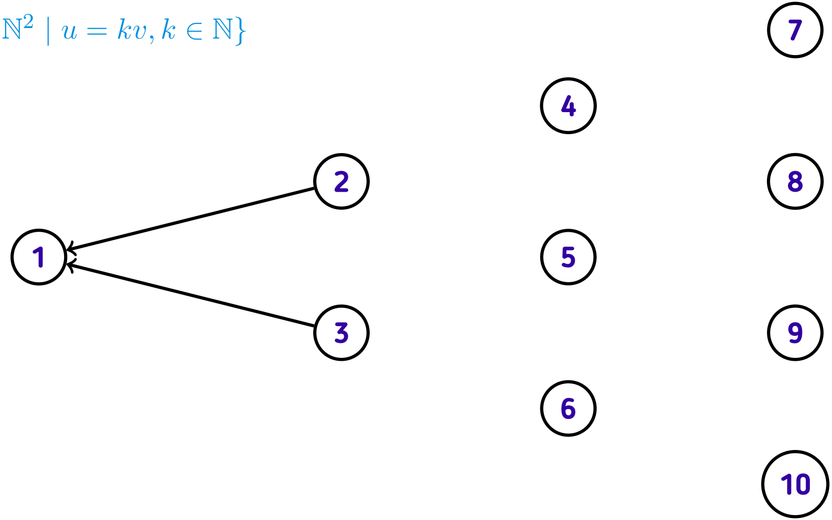
- Os vértices e as arestas são definidas por relações
- Adequada para grafos complexos ou com infinitos vértices e arestas
- Os elementos do grafo são identificados sob demanda

$$G = G(V, E)$$
$$E = \{(u, v) \in \mathbb{N}^2 \mid u = kv, k \in \mathbb{N}\}$$

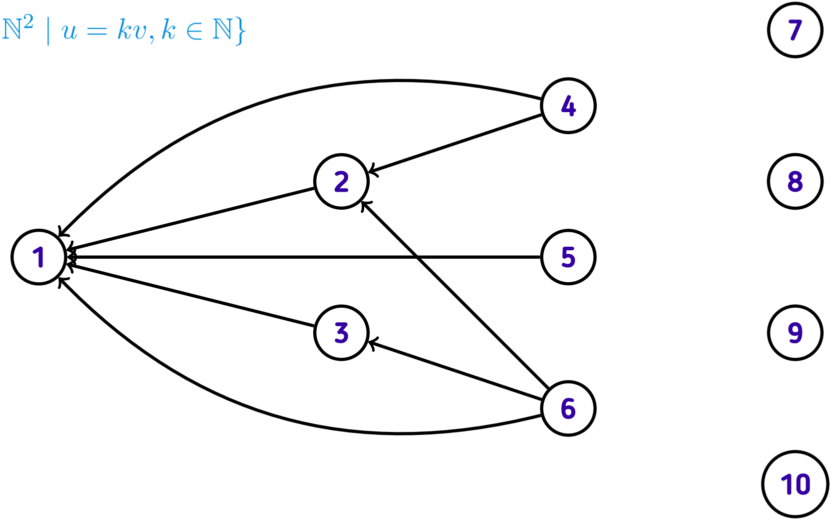
$$G = G(V, E)$$
$$E = \{(u, v) \in \mathbb{N}^2 \mid u = kv, k \in \mathbb{N}\}$$



$$G = G(V, E)$$
$$E = \{(u, v) \in \mathbb{N}^2 \mid u = kv, k \in \mathbb{N}\}$$

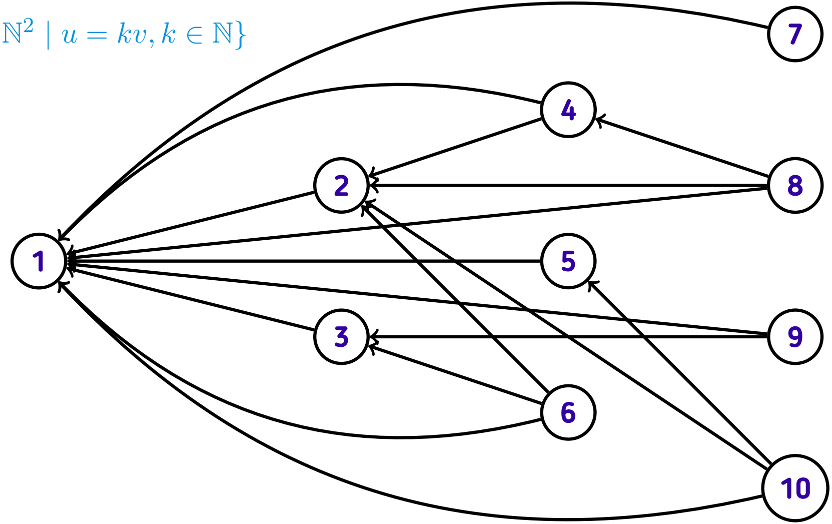


$$G = G(V, E)$$
$$E = \{(u, v) \in \mathbb{N}^2 \mid u = kv, k \in \mathbb{N}\}$$



$$G = G(V, E)$$

$$E = \{(u, v) \in \mathbb{N}^2 \mid u = kv, k \in \mathbb{N}\}$$



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.**