Projeto e Análise de Algoritmos
2024.2

Grafos e Corretude de Algoritmos

Prof. Marcio Costa Santos DCC/UFMG

GRAFOS E REPRESENTAÇÃO

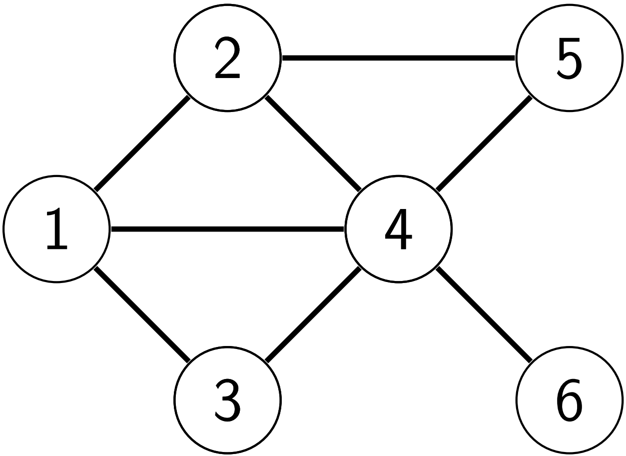
Grafos

- Grafo $G = (V, E)$.
- Vizinhança: $N(v) = \{u | vu \in E\}$.
- Grau $d(v) = |N(v)|$.
- Caminho conjunto de arestas $(v_1 v_2, , v_{k-1} v_k)$ que não passa duas vezes menos vértice.
- Ciclo conjunto de arestas $(v_1 v_2, , v_k v_1)$ que não passa duas vezes menos vértice, exceto o v_1.

Grafos

- Subgrafo H é subgrafo de G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$.
- Subgrafo Induzido se H é subgrafo de G e o conjunto de arestas em $V(H)$ é igual em H e G.
- Conexo: se existe um caminho entre todo par de vértice.

Representação



Grafos - Representação

- Representação:
  - Lista de Adjacência.
  - Matriz de Adjacência($\star$).

Representação - Matriz de Adjacência

	1	2	3	4	5	6
1	0	1	1	1	0	0
2	1	0	0	1	1	0
3	1	0	0	1	0	0
4	1	1	1	0	1	1
5	0	1	0	1	0	0
6	0	0	0	1	0	0

Representação - Lista de Adjacência

- $adj[1] = [2, 3, 4]$.
- $adj[2] = [1, 4, 5]$.
- $adj[3] = [1, 4]$.
- $adj[4] = [1, 2, 3, 5, 6]$.
- $adj[5] = [2, 4]$.
- $adj[6] = [4]$.

Vizinhança

- $V(G) = \{1, \ldots, n\}$.
- $E(G)$ é o conjunto de arestas.
- $N(v) = \{u \in V(G) \mid uv \in N(v)\}$.
- Lista de Adjacência: $O(1)$
- Matriz de Adjacência: $O(n)$

Vizinhança - Lista de Adjacência

Entrada: Grafo $G(V, E)$, vértice $v \in V(G)$
Saída: $N(v)$
retorna $adj[v]$;

Vizinhança - Matriz de Adjacência

Entrada: Grafo $G(V, E)$, vértice $v \in V(G)$
Saída: $N(v)$

$lst \leftarrow \emptyset$;
for *u de 1 até n* **do**
    if $M[v][u]$ **then**
        Inclui u em lst;
    end
end
retorna lst;

Grau

- $d(v) = |N(v)|$.
- Lista de Adjacência: $O(d(v))$
- Matriz de Adjacência: $O(n)$

Grau - Lista de Adjacência

Entrada: Grafo $G(V, E)$, vértice $v \in V(G)$
Saída: $d(v)$
$lst \leftarrow adj[v].head$;
$deg \leftarrow 0$;
while $lst \neq \lambda$ **do**
    $lst \leftarrow lst.next$;
    $deg \leftarrow deg + 1$;
end
retorna deg;

Grau - Matriz de Adjacência

Entrada: Grafo $G(V, E)$, vértice $v \in V(G)$
Saída: $d(v)$

$deg \leftarrow 0$;
for *u de 1 até n* **do**
    if $M[v][u]$ **then**
        $deg \leftarrow deg + 1$;
    end
end
retorna deg;

Pertinência de Aresta

- Responder se $uv \in E(G)$?
- Lista de Adjacência: $O(d(v))$
- Matriz de Adjacência: $O(1)$

Pertinência de Aresta - Lista de Adjacência

Entrada: Grafo $G(V, E)$, vértice $v, u \in V(G)$
Saída: $uv \in E(G)$?
$lst \leftarrow adj[v].head$;
while $lst \neq \lambda$ **do**
    if $lst = u$ **then**
        retorna *true*
    end
    $lst \leftarrow lst.next$;
end
retorna *false*;

Pertinência de Aresta - Matriz de Adjacência

Entrada: Grafo $G(V, E)$, vértice $v, u \in V(G)$
Saída: $uv \in E(G)$?
retorna $M[v][u]$

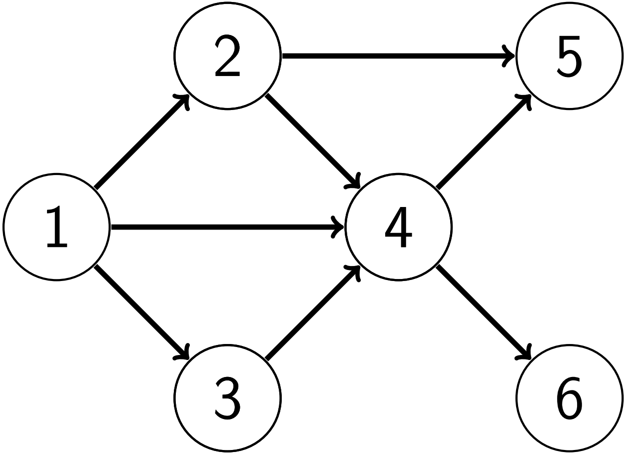
Grafos Direcionados

- Grafo direcionado $G = (V, A)$.
- Vizinhança de saída $N^+(v) = \{u|vu \in A\}$
- Vizinhança de entrada $N^-(v) = \{u|vu \in A\}$
- Grau de saída $d^+(v) = |N^+(v)|$
- Grau de entrada $d^-(v) = |N^-(v)|$
- Chamamos essas arestas de arcos

Grafos Direcionados

- Representação:
  - Lista de Adjacência.
  - Matriz de Adjacência($\star$).
- Cuidado que os arcos possuem sentidos!

Representação



CORRETUDE DE ALGORITMOS

Corretude de Algoritmos

- É necessário mostrar que nossos algoritmos estão corretos.
- Provas matemáticas formais.
- **indução matemática**.
- **invariantes de laço**.

Invariantes de Laço

- É uma relação/propriedade entre as variáveis do algoritmo.
- É verdade antes da execução do laço.
- É verdade depois de cada execução do laço.

Soma de elementos num vetor

Entrada: vetor de inteiros x, tamanho n de x.

$soma \leftarrow 0$;

para $i = 1$ *até* n **faça**

  $soma \leftarrow soma + x[i]$;

fim

retorna $soma$;

Soma de elementos num vetor - Invariantes

Entrada: vetor de inteiros x, tamanho n de x.

$soma \leftarrow 0$;

para $i = 1$ *até* n **faça**

  $soma \leftarrow soma + x[i]$;

fim

retorna $soma$;

Invariante

`soma` é a soma de todos os elementos do vetor, até aquela iteração. $soma = \sum_{i=1}^{n} x[i]$

Maior elementos num vetor

Entrada: vetor de inteiros x, tamanho n de x.
$max \leftarrow x[1]$;
para $i = 2$ *até* n **faça**
    if $x[i] > max$ **then**
        $max \leftarrow x[i]$;
    end
fim
retorna max;

Maior elementos num vetor - Invariantes

Entrada: vetor de inteiros x, tamanho n de x.

$max \leftarrow x[1]$;
para $i = 2$ *até* n **faça**
    if $x[i] > max$ **then**
        $max \leftarrow x[i]$;
    end
fim
retorna max;

Invariante

`max` é o maior de todos os elementos do vetor, até aquela iteração. $max = \max_{j \leq i} x[j]$

Fatorial

Entrada: inteiro n.
if $n \geq 2$ **then**
    retorna $n.fat(n-1)$;
end
else
    retorna 1;
end

Algoritmo 1: fat(n)

Indução Matemática

- Uma forma mais formal e geral.
- Estrutura de uma prova por indução:

Base:	valor inicial, normalmente 0 ou 1.
Hipótese:	assumimos que é verdade para k.
Passo:	provamos que se vale para k também vale para $k+1$.