Representação computacional
Estrutura de dados – Grafo

Matriz de adjacências

Matriz de incidências

Lista de adjacências

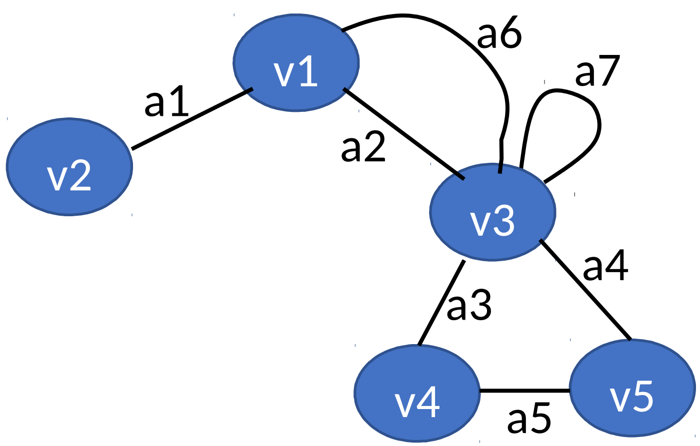
Representação computacional
Estrutura de dados – Grafo

Matriz de adjacências

Para fins de implementação dos algoritmos é necessária uma representação do grafo em sua estrutura de dados capaz de ser instanciada na memória do computador. Uma das formas de representação é a Matriz de adjacências:

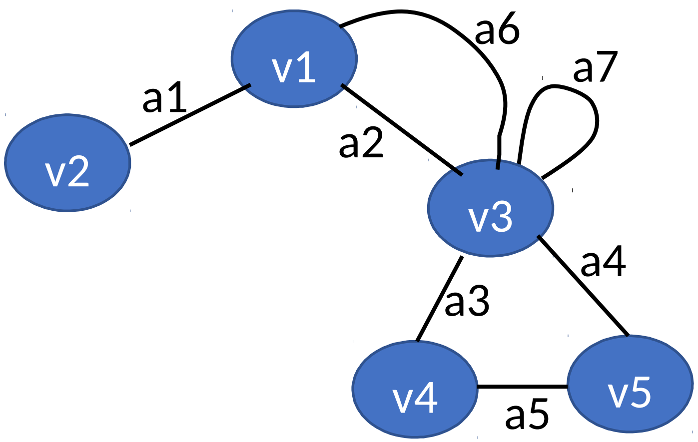
Suponha G(V,A), V={v1,v2,v3,...,vn}, podemos formar uma matriz n × n em que o elemento i, j é o número de arestas entre os vértices v_i e v_j. Essa matriz é chamada de matriz de adjacências do grafo em relação à ordem dos vértices:

a_{ij} = p, se existirem p arcos entre v_i e v_j.

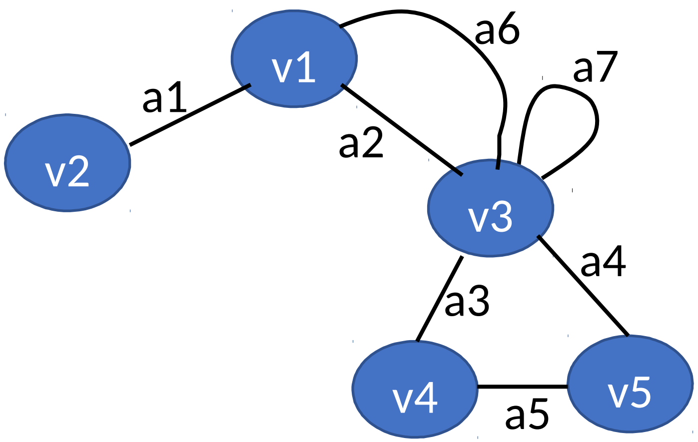


	v1	v2	v3	v4	v5
v1					
v2					
v3					
v4					
v5					

a_{ij} = p, se existirem p arcos entre v_i e vj.



	v1	v2	v3	v4	v5
v1	0	1	2	0	0
v2	1	0	0	0	0
v3	2	0	2	1	1
v4	0	0	1	0	1
v5	0	0	1	1	0



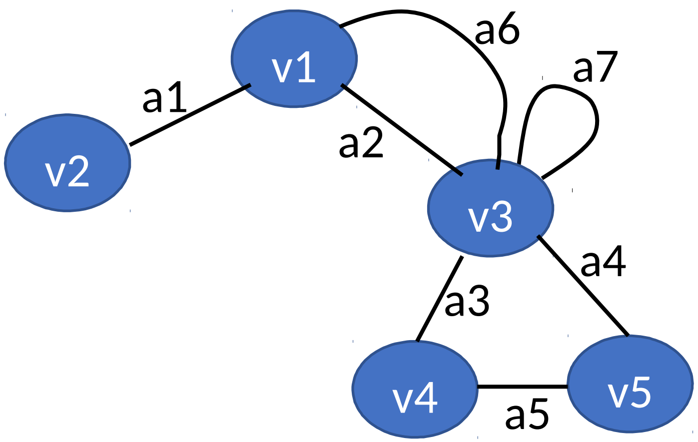
	v1	v2	v3	v4	v5
v1	0	1	2	0	0
v2	1	0	0	0	0
v3	2	0	2	1	1
v4	0	0	1	0	1
v5	0	0	1	1	0

Simetria em relação à diagonal principal: $a_{ii} \neq 0$ então v_i possui laço

Grau(v_i)=Somatório da i-ésima linha (ou coluna);

Para $i \neq j$, arestas múltiplas ocorrem quando $a_{ij} > 1$ na i-ésima linha

a_{ij} = p, se existirem p arcos entre v_i e v_j.



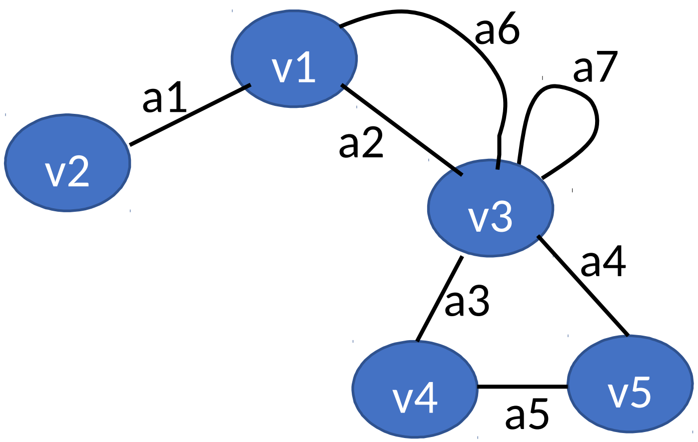
G(V,A)
V={v1,v2,v3,v4,v5}
A={a1,a2,a3,a4,a5,a6,a7}

Lista de adjacências:
a1=(1,2); a2=(1,3); a3=(3,4);a4=(3,5);
a5=(4,5); a6=(1,3); a7=(3,3)

Entrada de dados:
número de vértices e lista de adjacências (em um vetor, por exemplo);

A entrada pode ser feita em um arquivo texto:
5 #vértices
1 2, 1 3, 3 4, 3 5, 4 5, 1 3, 3 3

a_{ij} = p, se existirem p arcos entre v_i e v_j.



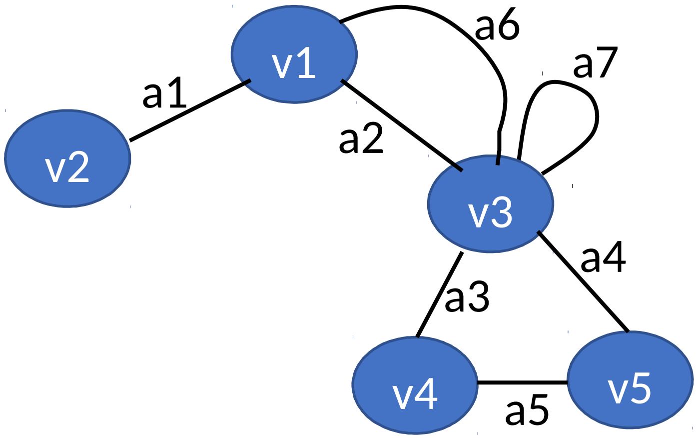
G(V,A)
V={v1,v2,v3,v4,v5}
A={a1,a2,a3,a4,a5,a6,a7}

Lista de adjacências:
a1=(1,2); a2=(1,3); a3=(3,4);a4=(3,5);
a5=(4,5); a6=(1,3); a7=(3,3)

Implemente a matriz de adjacências para representar um grafo não orientado, apresentando as seguintes funcionalidades ao usuário:

1. determinação dos nós com arestas em laço;
2. determinação do par de nós com arestas múltiplas e a quantidade dessas arestas entre esse par de nós;
3. Determinação do grau de um vértice;
4. Identificação de vértices isolados;
5. Verificar se o número de arestas é no máximo igual a $\binom{n}{2} = \frac{(n(n-1))}{2}$ onde n=|V|

Segue...



G(V,A)
V={v1,v2,v3,v4,v5}
A={a1,a2,a3,a4,a5,a6,a7}

Lista de adjacências:
a1=(1,2); a2=(1,3); a3=(3,4);a4=(3,5);
a5=(4,5); a6=(1,3); a7=(3,3)

7. Verificar pela matriz, se:

a) $$\sum_{v \in V} grau(v) = 2|E|$$

b) Se o número total de vértices com grau ímpar é sempre par

Representação computacional
Estrutura de dados – Grafo

Matriz de incidências

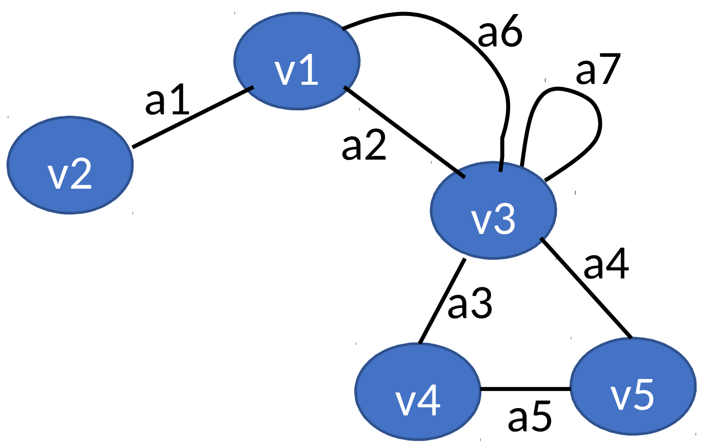
G(V,A)
V={v1,v2,v3,v4,v5}
A={a1,a2,a3,a4,a5,a6,a7}

grau(v_i)=somatório na i-ésima linha;

$a_{ij} > 1 \rightarrow$ laço;

Colunas iguais→ arestas múltiplas;

Como identificar a adjacência de um vértice? Ex: v1 é adjacente a quais vértices?



	a1	a2	a3	a4	a5	a6	a7
v1	1	1	0	0	0	1	0
v2	1	0	0	0	0	0	0
v3	0	1	1	1	0	1	2
v4	0	0	1	0	1	0	0
v5	0	0	0	1	1	0	0

Para um grafo (V,E)
|V|=n
|E|=m

Em termos de complexidade espacial, considerando o grafo completo como o pior caso ambas as matrizes de adjacências e incidências ocupam um espaço bidimensional de tamanho *total_linhas* * *total_colunas*:

- A matriz de adjacências é esparsa e ocupa um espaço sempre correspondente a exatamente $n*n = n^2$ (independente do pior ou melhor caso)

- A matriz de incidências ocupa um espaço n * m.

No pior caso, $m=n(n-1)/2 \rightarrow n * m = n * (n(n-1)/2)$

O seja, no pior caso, a matriz de incidências demanda um espaço da ordem de n^3.

Representação computacional
Estrutura de dados – Grafo

Lista de adjacências

A matriz de adjacências independentemente do caso (melhor ou pior) sempre demanda espaço de memória da ordem n^2.
A matriz de incidências, no pior caso, pode alcançar um consumo de memória de ordem n^3.
Ambas matrizes (adjacências e incidência) podem ser bastante esparsas

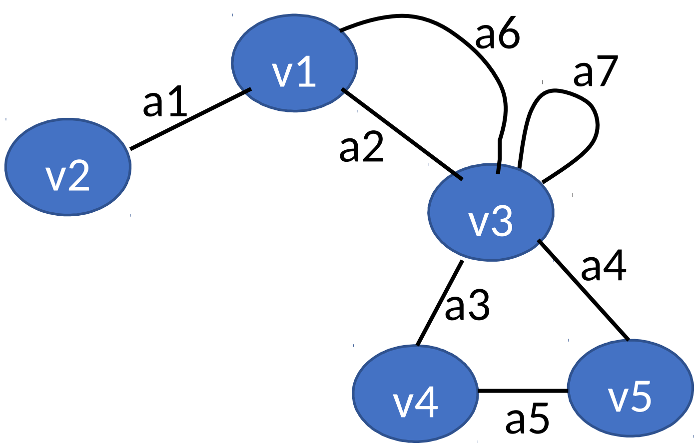
Por sua vez, a lista de adjacências "enxuga" a representação modelando apenas as conexões que de fato existam no grafo, não sendo uma estrutura esparsa.

A lista de adjacências consome um espaço da ordem n+2m.

Ainda que seja assim, no pior caso, a lista de adjacências consome espaço:

$$n+2m = n+(n(n-1)) = n+n^2-n = n^2$$

Ou seja, uma estrutura de dimensões da ordem de n^2

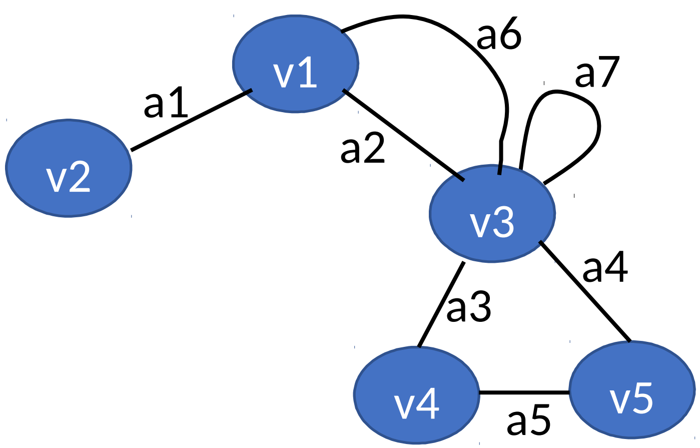


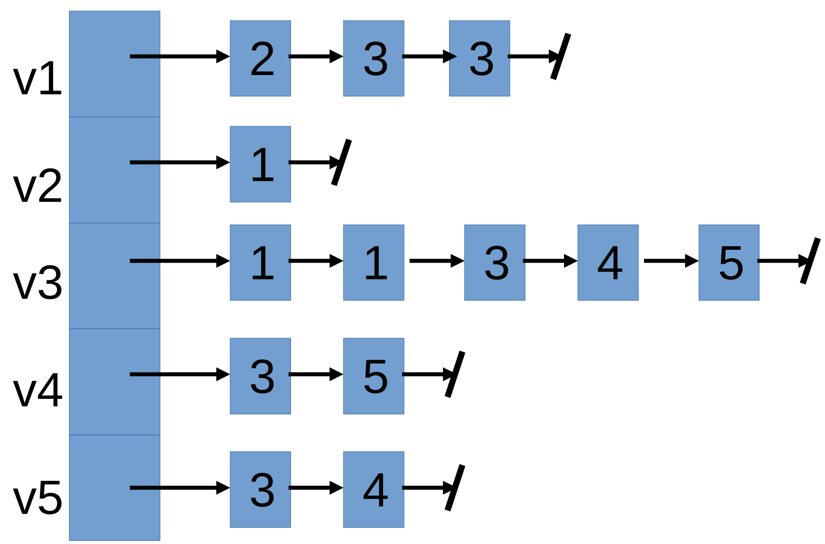
G(V,A)
V={v1,v2,v3,v4,v5}
A={a1,a2,a3,a4,a5,a6,a7}

No pior caso a lista de adjacências consome espaço da ordem de n+2m

A lista de adjacências corresponde a por um vetor V de dimensão n.
Cada elemento de V contém dois campos: a identificação de um vértice e um ponteiro para uma lista encadeada contendo os vizinhos do vértice Correspondente.

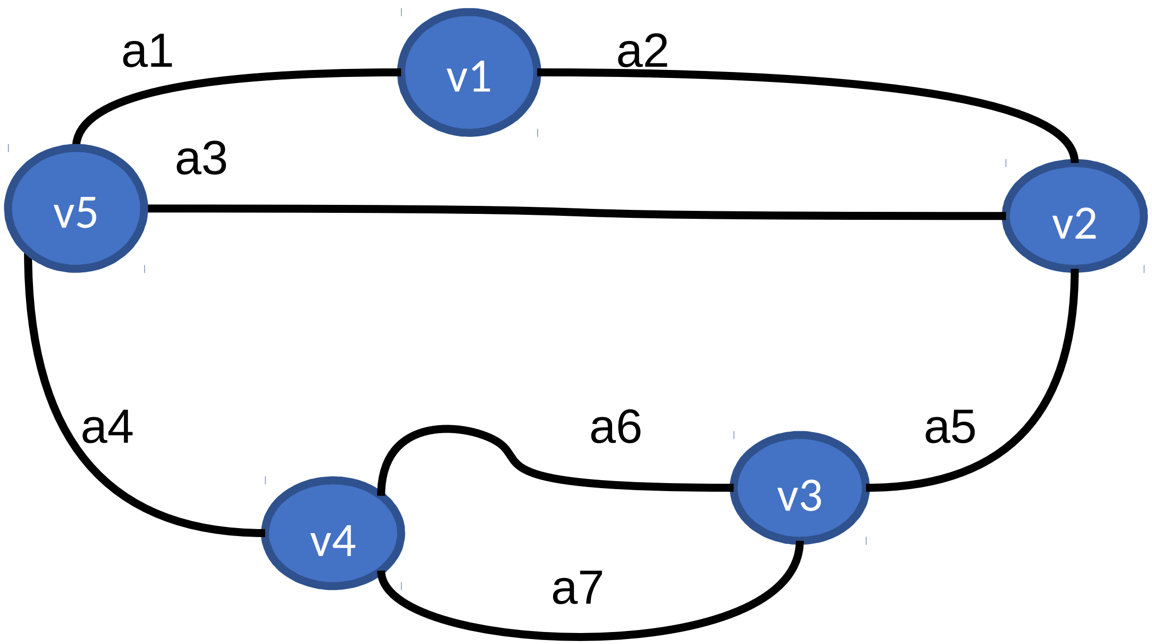
A implementação não indexada (lista encadeada) é um fator que incrementa a Complexidade temporal da implementação em lista de adjacências.





Exercícios:

1)



Escreva a matriz de adjacência de incidência e a lista de adjacências para o grafo acima.

Exercícios:

2) Desenhe o grafo representado pela a matriz de adjacência abaixo:

0	1	1	2	0
1	0	0	0	1
1	0	0	1	1
2	0	1	0	0
0	1	1	0	0

Exercícios:

3) Desenhe o grafo representado pela a matriz de incidência abaixo:

0	1	1	1	1	1	1	0
0	1	0	1	0	0	0	1
0	0	0	0	0	0	0	1
1	0	1	0	1	0	1	0
1	1	0	0	0	1	0	0

As estruturas vistas aqui podem ser adaptadas à representação de digrafos (grafos direcionados);

Será tratado oportunamente...