

Algoritmos e Estruturas de Dados II

Grafos – tipo abstrato de dados

Thiago A. S. Pardo

Profa. M. Cristina

Material de aula da Profa.

Josiane M. Bueno

Grafos

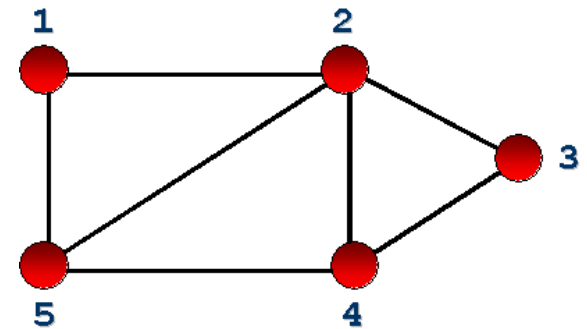
Estruturas de Dados

- Como vimos na última aula...
 - Há duas representações usuais:
 - **Matriz de Adjacências**
 - **Listas de Adjacências**

Grafos

Matriz de Adjacências

- Exemplo



$M =$

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

vértices

Grafos

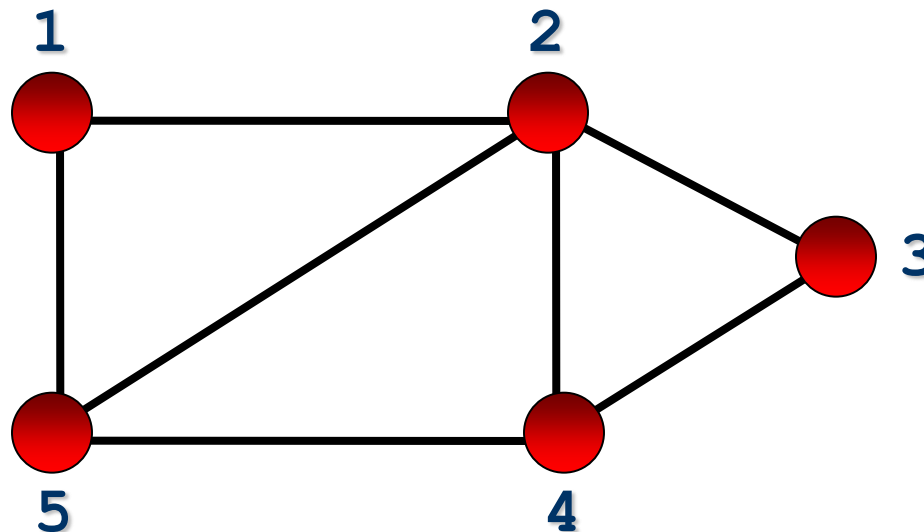
Listas de Adjacências

- Dado um grafo $G = (V, E)$, as **listas de adjacências** A é um conjunto de $|V|$ listas $A(v)$, uma para cada vértice v pertencente a V
- Cada lista $A(v)$ é denominada **lista de adjacências** do vértice v e contém os vértices w adjacentes a v em G
- Ou seja, as **listas de adjacências** consistem um vetor de $|V|$ elementos que são capazes de apontar, cada um, para uma lista linear
 - O i -ésimo elemento do vetor aponta para a lista linear das arestas que são adjacentes ao vértice i

Grafos

Listas de Adjacências

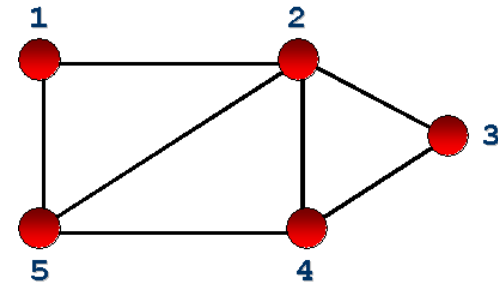
- Como são as listas de adjacências do grafo a seguir?



Grafos

Listas de Adjacências

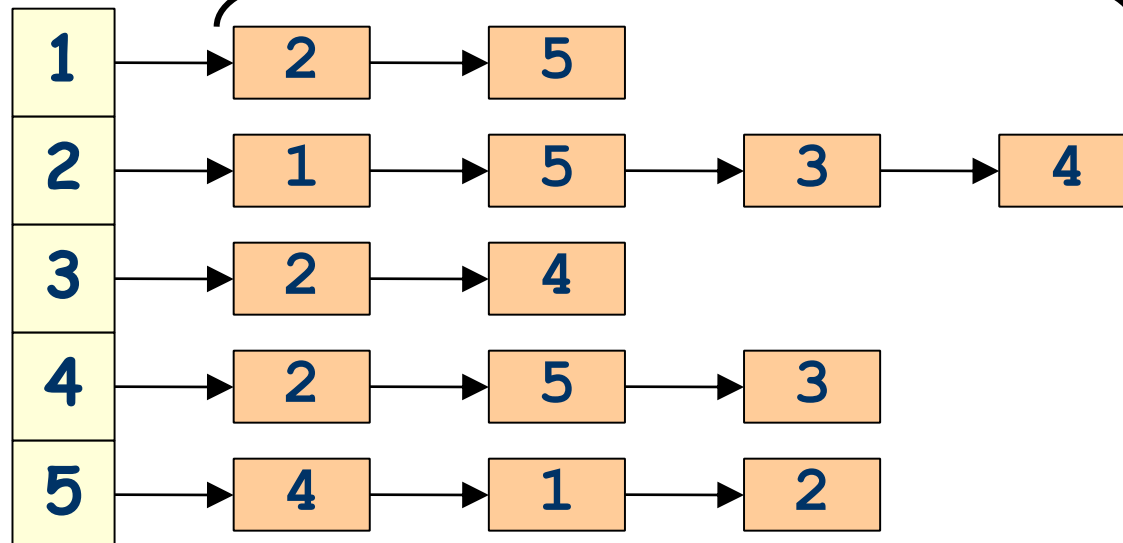
- Possível resposta:



vetor



Listas lineares



Grafos

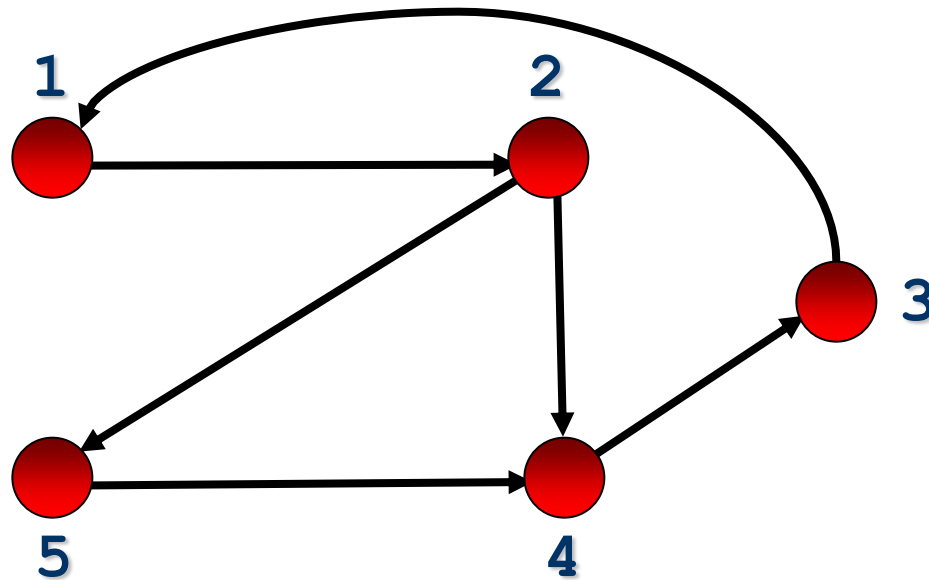
Listas de Adjacências

- Em grafos não direcionados, cada aresta é representada 2 vezes

Grafos

Listas de Adjacências

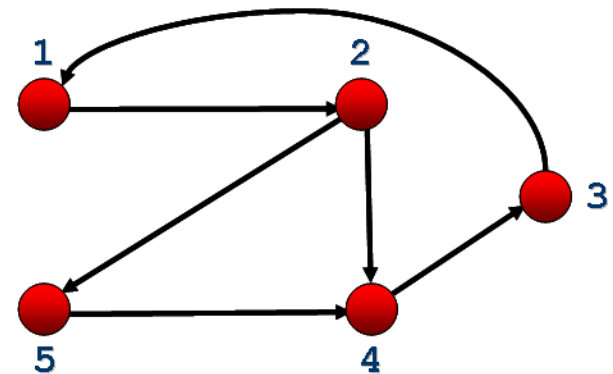
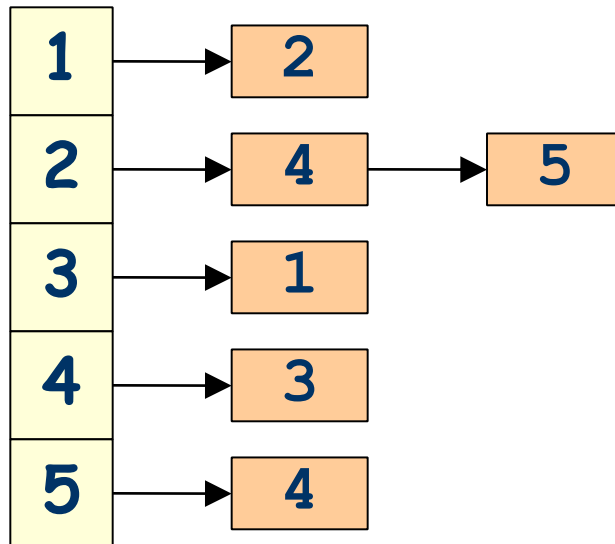
- Como representar o dígrafo abaixo?



Grafos

Listas de Adjacências

- Possível resposta:

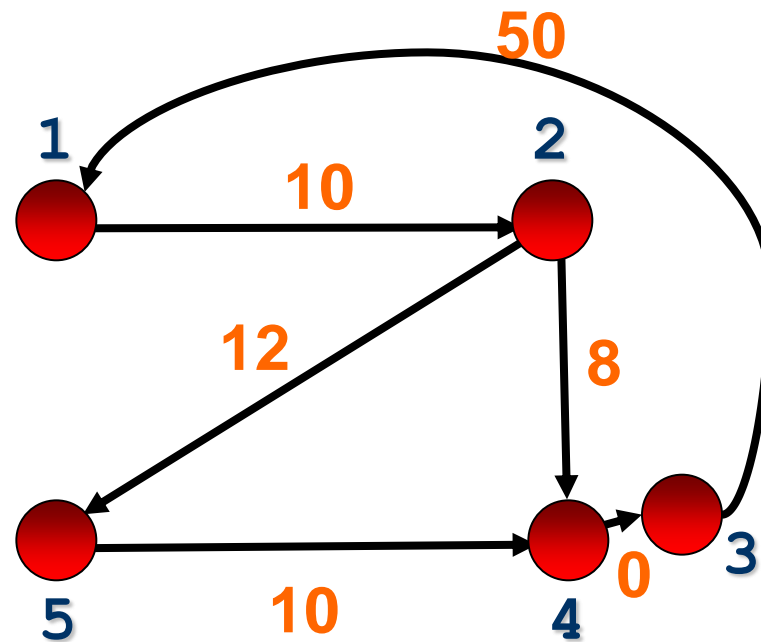


Vértice i aponta para
vértices adjacentes
→ Outras opções?

Grafos

Listas de Adjacências

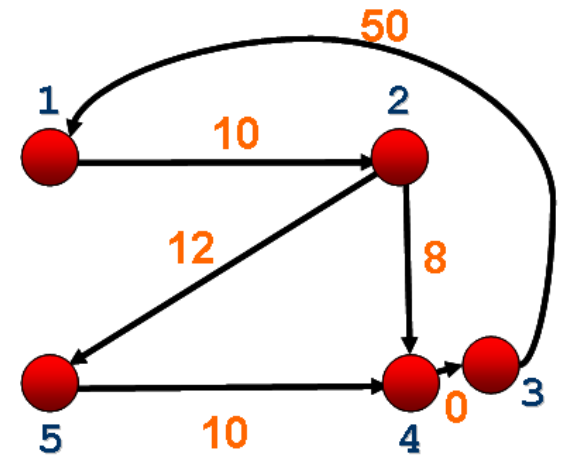
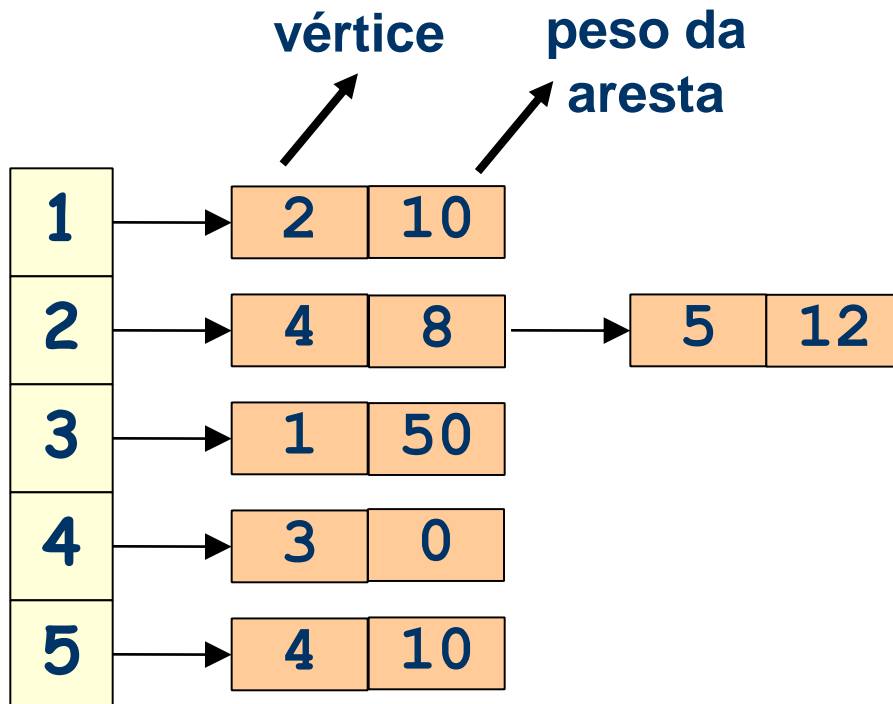
- Como representar o grafo direcionado e valorado abaixo?



Grafos

Listas de Adjacências

- Possível resposta:



Grafos

Listas de Adjacências

- **Maior complexidade na representação**
- **Propriedades**
 - Armazenamento: ?
 - Teste se aresta (i,j) está no grafo: ?

Grafos

Listas de Adjacências

- **Maior complexidade na representação**
- **Propriedades**
 - Armazenamento: $O(|V| + |E|)$
 - Teste se aresta (i,j) está no grafo: $O(d_i)$, com d_i sendo o grau do vértice i

Grafos

Listas de Adjacências

- Boa representação para **grafos esparsos**, em que $|E|$ é muito menor do que $|V|^2$
- Representação **compacta** e usualmente utilizada na **maioria das aplicações**
- Desvantagem: tempo $O(|V|)$ para determinar se existe uma aresta entre i e j , pois podem haver $|V|$ elementos na lista de adjacências de i

Grafos

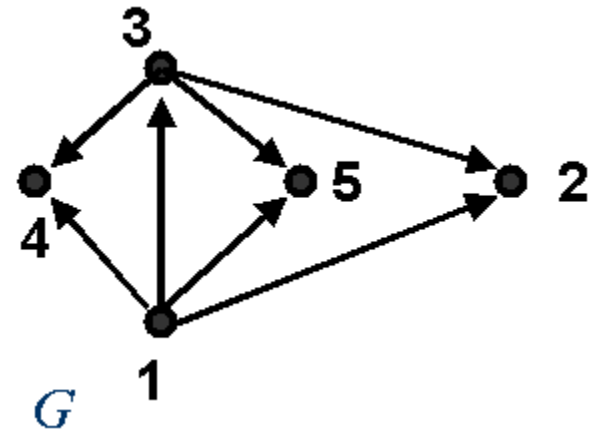
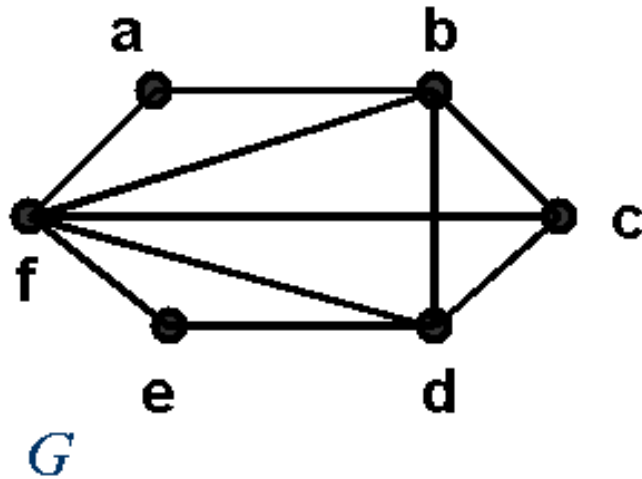
Listas de Adjacências

- **Atenção**

- Os vértices adjacentes a um vértice i podem ser armazenados na lista de adjacências de i em ordem arbitrária ou não
- Não é incomum a lista de adjacências não ser organizada por “adjacências arbitrárias”, mas por **divergências** ou **convergências** das arestas
- Como em qualquer lista, há liberdade para haver variações na representação (nós cabeça, sentinelas, etc.)

Grafos

Exercício de Fixação

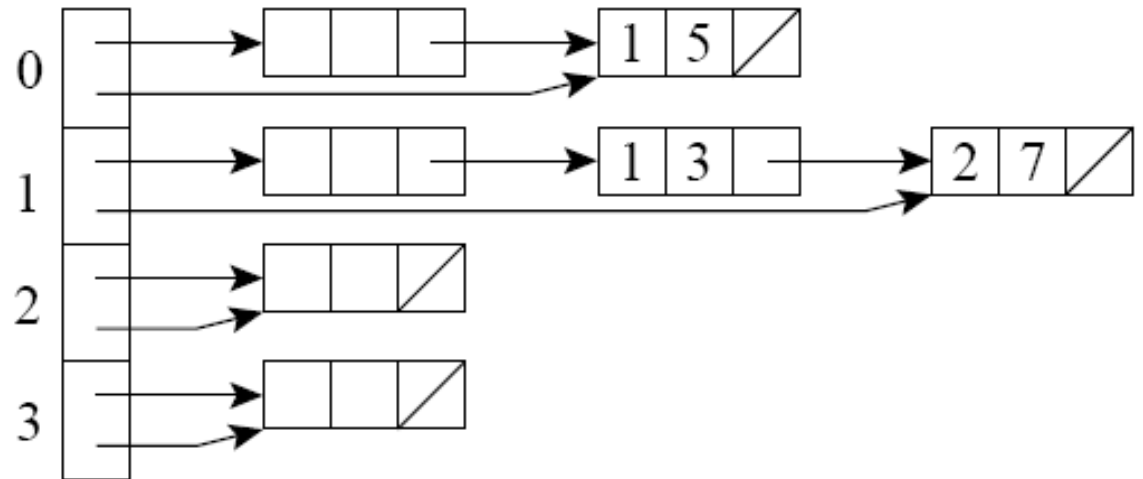
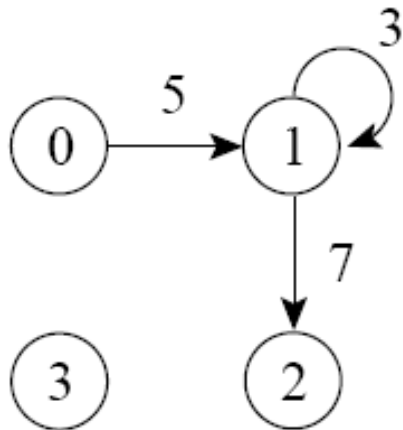


- Represente os grafos acima utilizando listas de adjacências

Grafos

Listas de Adjacências

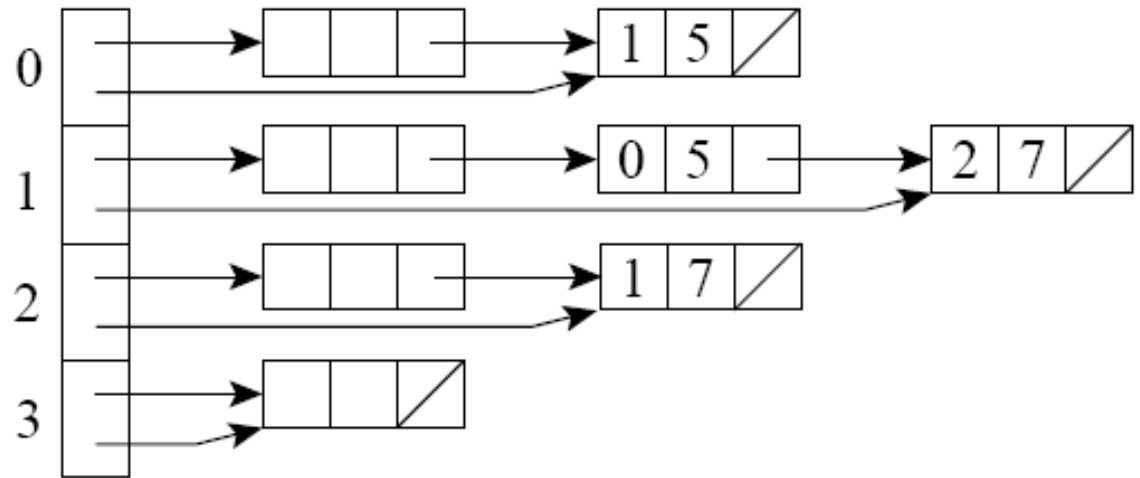
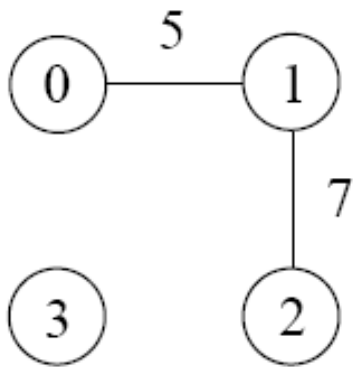
- Representação com nós cabeça



Grafos

Listas de Adjacências

- Representação com nós cabeça



Grafos

Listas de Adjacências

- **Implementação de algumas das operações mais comuns**
 - Criar grafo vazio
 - Inserir aresta
 - Retirar aresta
 - Existe aresta?
 - Obter lista de vértices adjacentes a um determinado vértice
 - Lista está vazia?
 - Retornar primeiro vértice da lista
 - Retornar próximo vértice adjacente da lista
 - Liberar memória utilizada pelo grafo
 - Imprimir grafo

Grafos

Listas de Adjacências

- **Exercício**

- Implementar sub-rotina que encontre o vértice adjacente a x com aresta de menor peso em um grafo valorado e direcionado

Grafos

Comparação

Comparação	Vencedor
Rapidez para saber se (x, y) está no grafo	Matriz de adjacências
Rapidez para determinar o grau de um vértice	Listas de adjacências
Menor memória em grafos pequenos	Listas: $(V + E)$ Matriz: $ V ^2$

Grafos

Comparação

Comparação	Vencedor
Menor memória em grafos grandes	Matriz de adjacências
Inserção/remoção de arestas	Matriz: $O(1)$ Listas: $O(d)$
Melhor na maioria dos problemas	Listas de adjacências
Rapidez para percorrer o grafo	Listas: $O(V + E)$ Matriz: $O(V ^2)$

Grafos

Estruturas de Dados

- Outra possibilidade consiste implementar grafos simulando **listas de adjacências em arranjos**
 - Como seria a estrutura?
 - E as operações?