

SCC0216 Modelagem Computacional em Grafos

**Thiago A. S. Pardo
Maria Cristina F. Oliveira**



Grafos

Estruturas de Dados

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

Grafos

Estruturas de Dados

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

Grafos

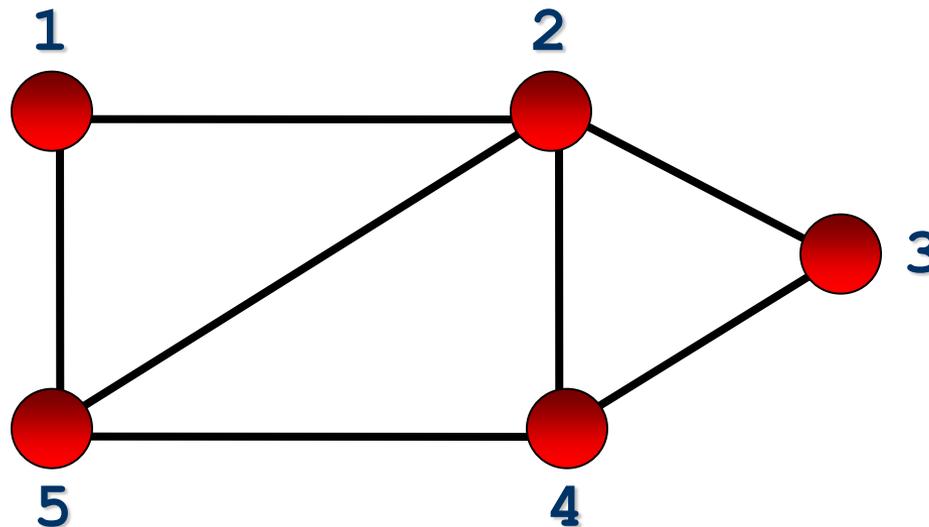
Listas de Adjacências

- Dado um grafo $G = (V, A)$, **listas de adjacências** L são um conjunto de $|V| = n$ listas $L(v)$, uma para cada vértice v pertencente a V
- Cada lista $L(v)$ 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 tradicionalmente em um vetor de $|V|$ elementos, em que cada elemento aponta 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

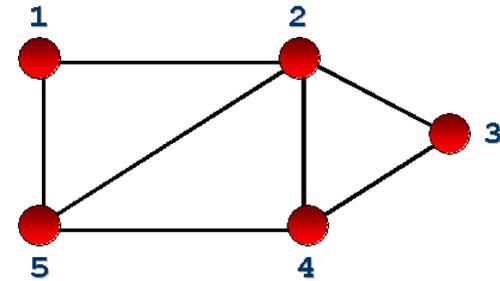
- Como são as listas de adjacências do grafo a seguir?



Grafos

Listas de Adjacências

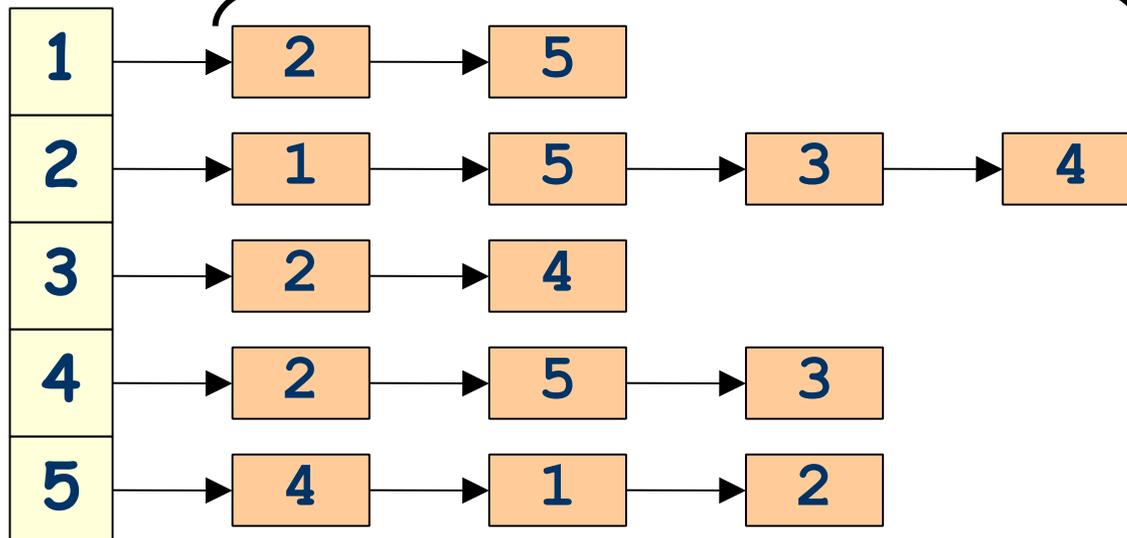
- Possível resposta



vetor



Listas lineares



Grafos

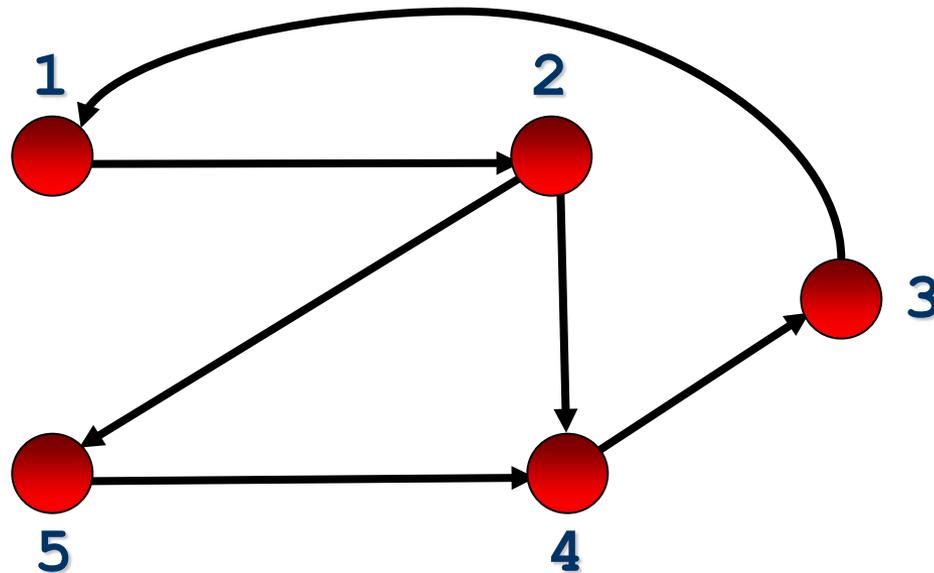
Listas de Adjacências

- **Atenção:** em grafos **não direcionados**, cada aresta é representada 2 vezes

Grafos

Listas de Adjacências

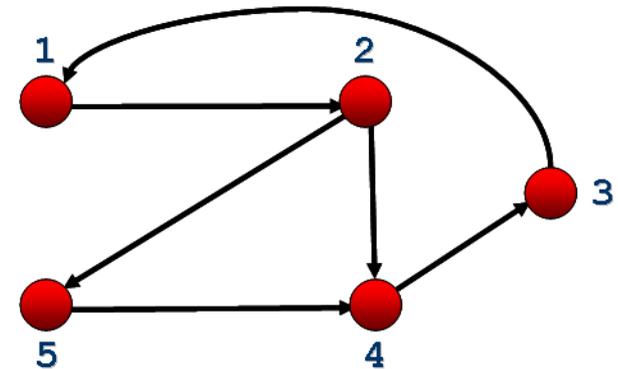
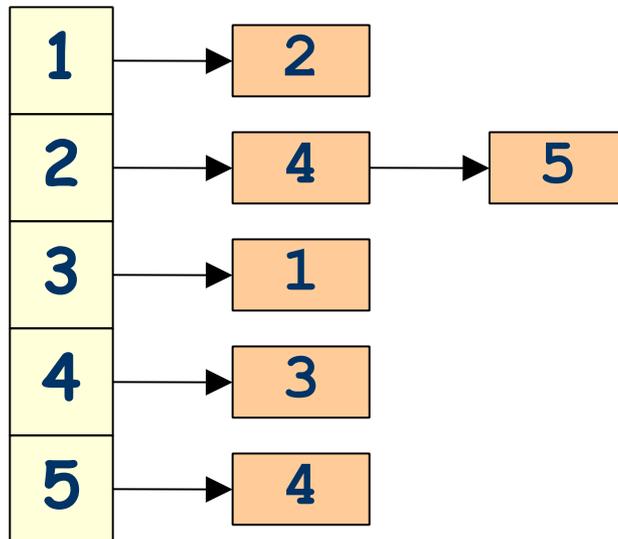
- Como representar o dígrafo abaixo?



Grafos

Listas de Adjacências

- Possível resposta

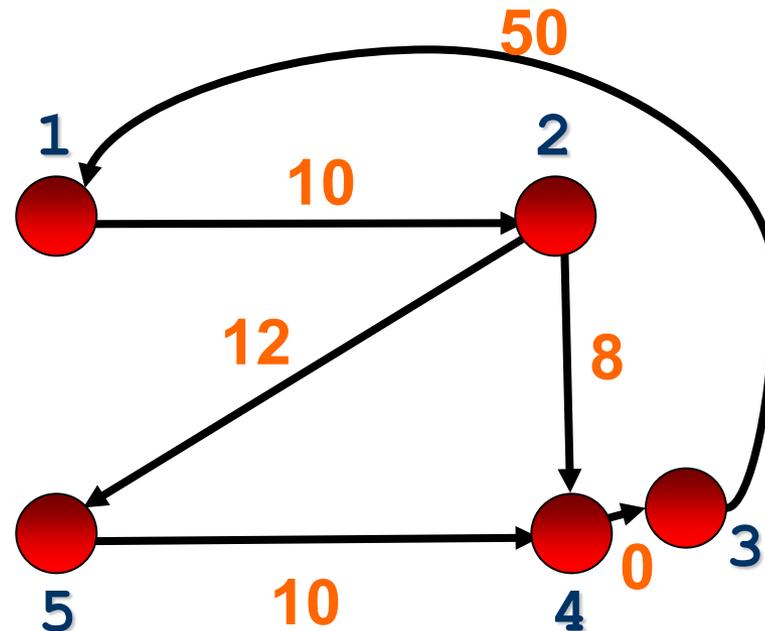


Vértice i aponta para
vértices adjacentes

Grafos

Listas de Adjacências

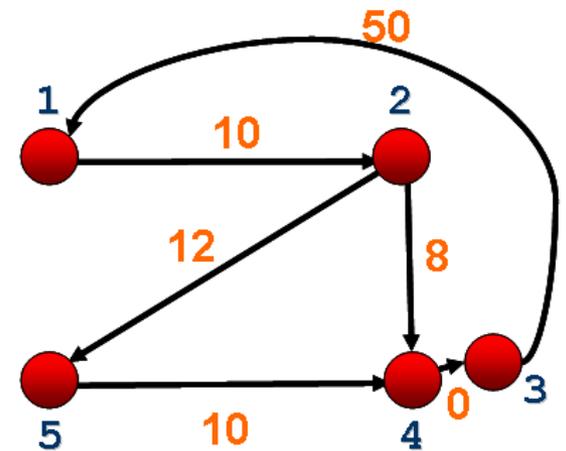
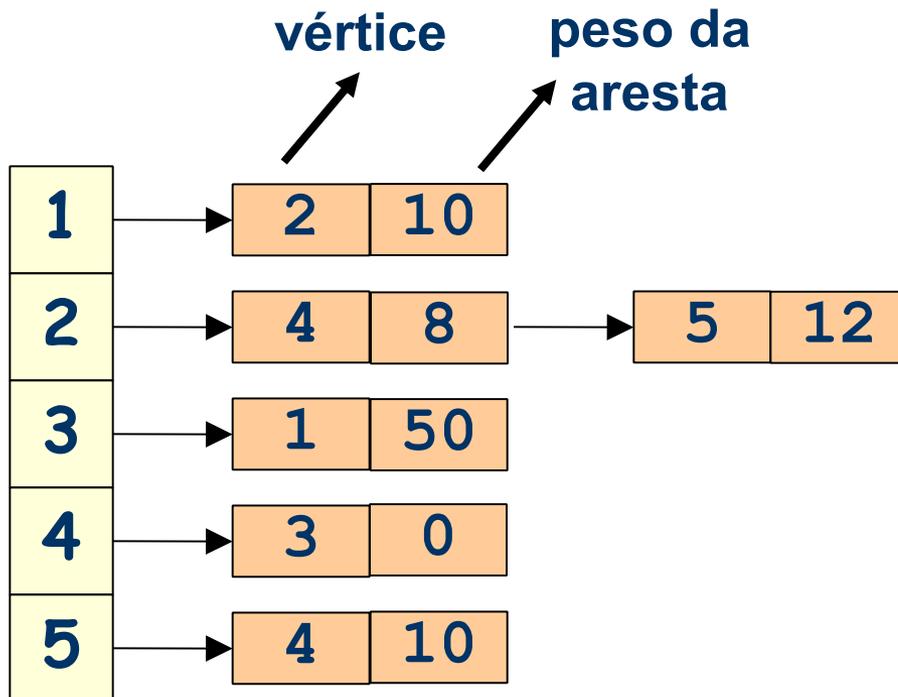
- Como representar o grafo direcionado e valorado abaixo?



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: $O(n + m)$, $n = |V|$, $m = |A|$
 - Teste se aresta (i,j) está no grafo: ?

Grafos

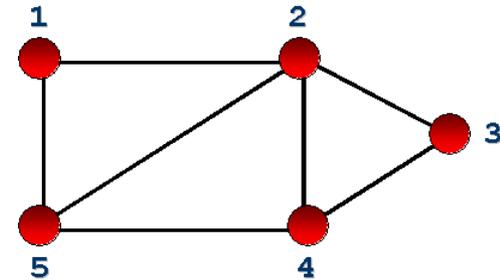
Listas de Adjacências

- **Maior complexidade na representação**
- **Propriedades**
 - **Armazenamento: $O(n + m)$, $n = |V|$, $m = |A|$**
 - **Teste se aresta (i,j) está no grafo: ?**
 - **Qual o pior caso?**

Grafos

Listas de Adjacências

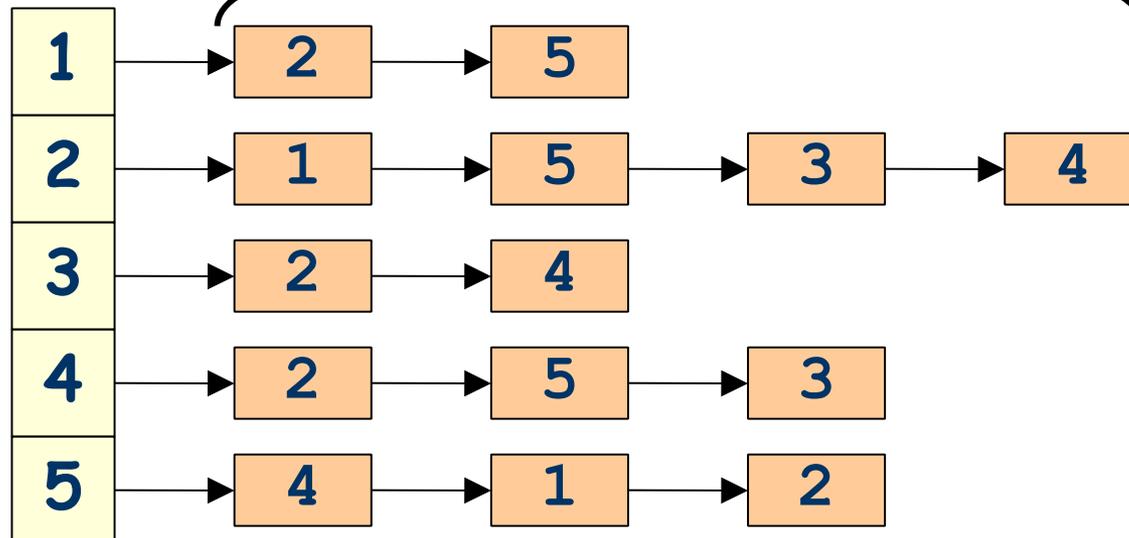
- Possível resposta



vetor



Listas lineares



Grafos

Listas de Adjacências

- **Maior complexidade na representação**
- **Propriedades**
 - Armazenamento: $O(|V| + |A|)$
 - 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 $|A|$ é muito menor do que $|V|^2$
- Representação **compacta** e usualmente utilizada na **maioria das aplicações**
- **Desvantagem**: tempo para determinar se existe uma aresta entre i e j é $O(|V|)$, pois a lista de adjacências de i pode ter $|V|$ elementos

Grafos

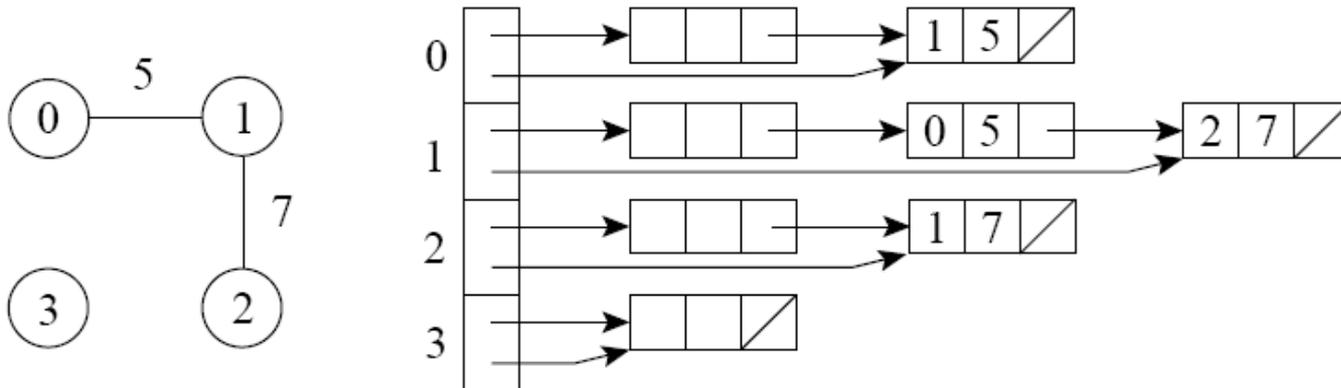
Listas de Adjacências

- **Atenção**
 - Os vértices adjacentes a um vértice i podem ser armazenados na lista de adjacências de i de **forma ordenada** ou **não**
 - Como em qualquer estrutura de dados, há liberdade para introduzir **variações** na representação (**nós de cabeçalho, outra lista linear em vez do vetor, etc.**)

Grafos

Listas de Adjacências

- Representação usual

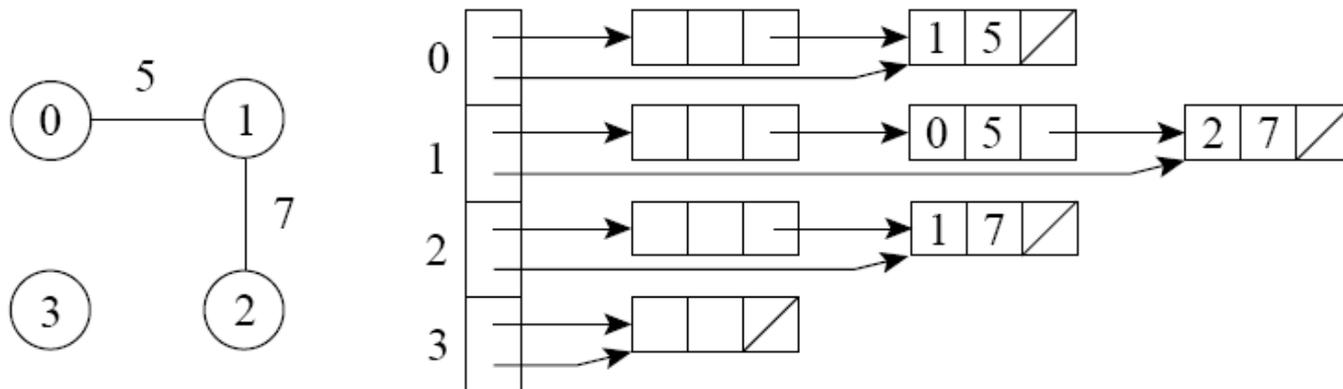


Questão: nós de cabeçalho são úteis?

Grafos

Listas de Adjacências

- Representação usual



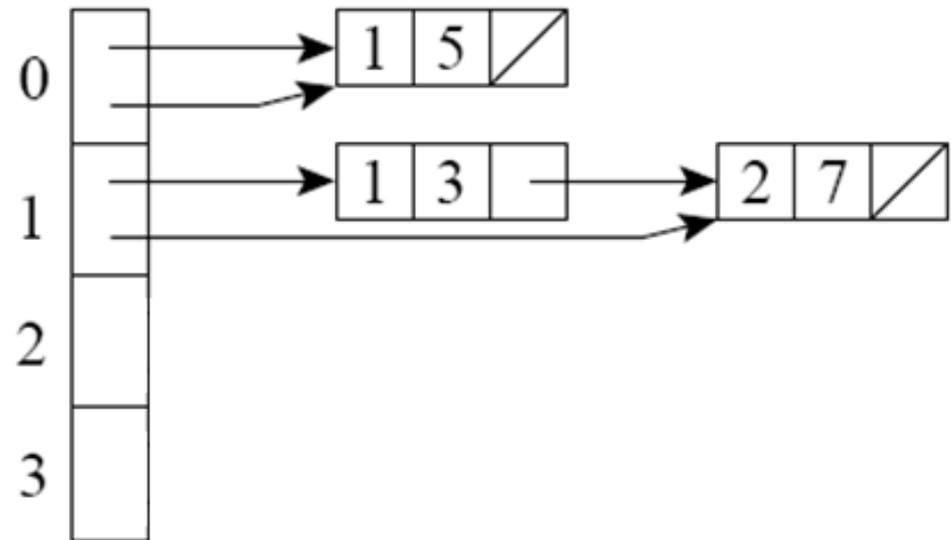
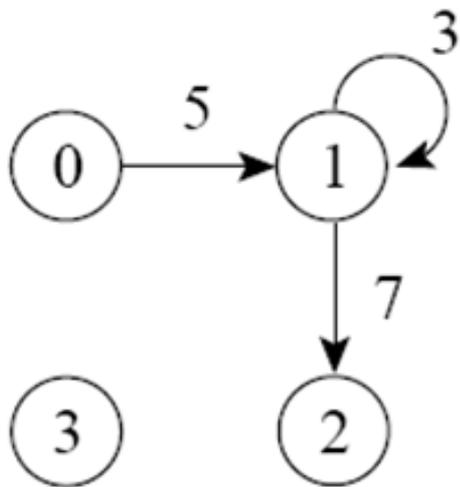
Questão: nós de cabeçalho são úteis?

- “Congelam-se” as estruturas dos vértices e das arestas, mexendo-se somente nos nós de cabeçalho, tornando o TAD mais genérico e facilmente adaptável para situações diferentes
 - Para registro do número de vizinhos, indicação de onde se parou algum processo, evitar checagem de endereço inválido, etc.

Grafos

Listas de Adjacências

- Representação usual

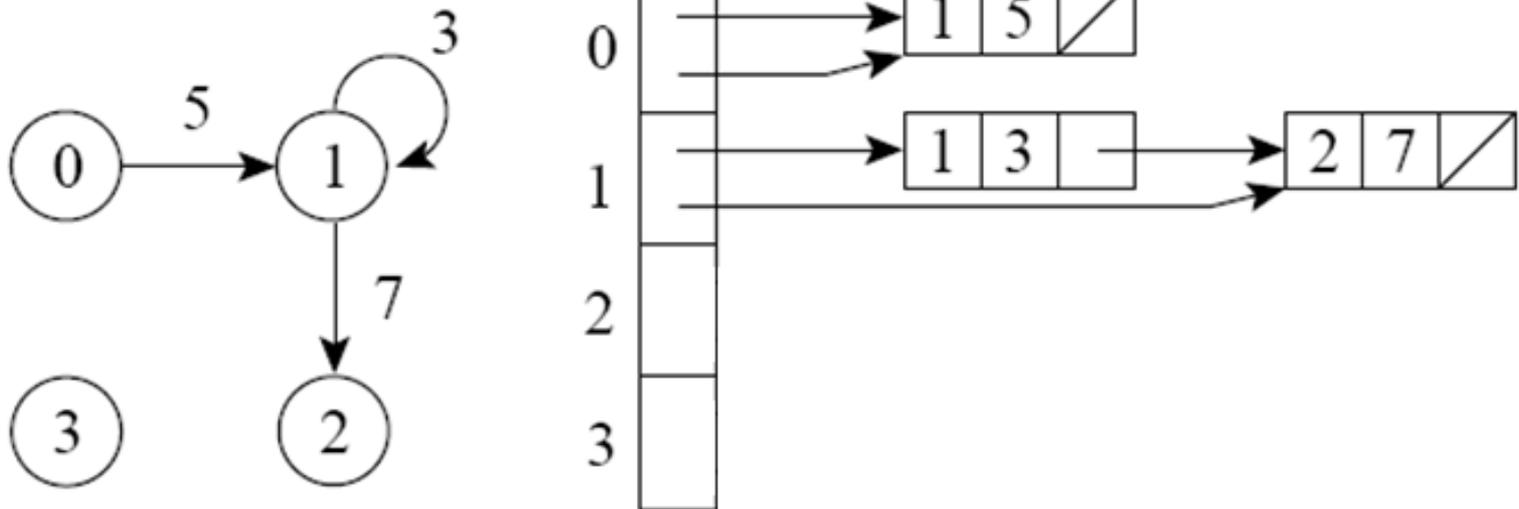


Questão: ponteiro para o último de cada lista?
Vantagens e desvantagens?

Grafos

Listas de Adjacências

- Representação usual



Questão: ponteiro para o último de cada lista?

Vantagens e desvantagens? Basicamente, estratégias diferentes de inserção e organização dos dados 😊, mas mais ponteiros ☹

Grafos

Comparação

Comparação	Vencedor
Rapidez para determinar se aresta (x,y) está no grafo	Matriz de adjacências
Rapidez para determinar o grau de um vértice	Listas de adjacências
Grafos esparsos	Listas de adjacências

Grafos

Comparação

Comparação	Vencedor
Grafos densos	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: $O(V + A)$ Matriz: $O(V ^2)$

Prática

TAD grafo



Grafos

Listas de Adjacências

- Declarar estrutura de dados das listas de adjacências (com nós de cabeçalho)

Grafos

Listas de Adjacências

- **Implementação de algumas das operações mais comuns (com nós de cabeçalho)**
 - Criar grafo vazio
 - Inserir aresta
 - Retirar aresta
 - Existe aresta?
 - Imprimir grafo
 - Liberar memória utilizada

Grafos

Listas de Adjacências

- Implementação de algumas das operações um pouco mais sofisticadas (com nós de cabeçalho)
 - Obter lista de vértices adjacentes a um determinado vértice
 - Lista de vértices adjacentes está vazia?
 - Retornar primeiro vértice da lista
 - Retornar próximo vértice adjacente da lista
 - Gerar grafo transposto (para dígrafos)

Prática

Desafios



Grafos

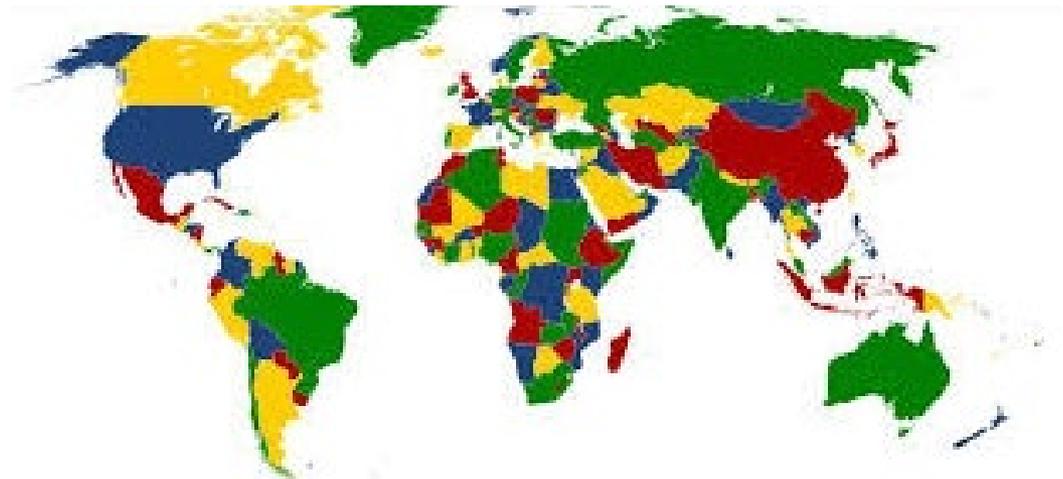
Listas de Adjacências

- Implementar função para contar o número de arestas de um grafo
 - Versão iterativa
 - Versão recursiva

Grafos

Listas de Adjacências

- Implementar função para colorir um mapa qualquer fornecido como entrada
 - Atenção: a coloração deve ser tal que países vizinhos devem ter cores diferentes e deve-se tentar minimizar o número de cores usadas



Grafos

Listas de Adjacências

- **Coloração de grafos**
 - Sugestão: algoritmo guloso de Welsh-Powell, proposto em 1975

