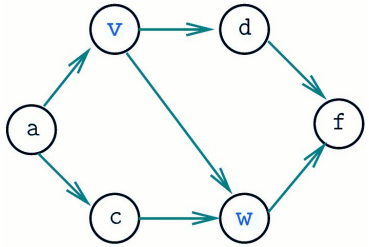
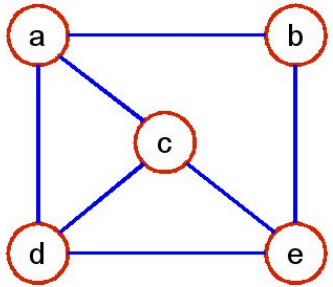
Estrutura de dados para grafo

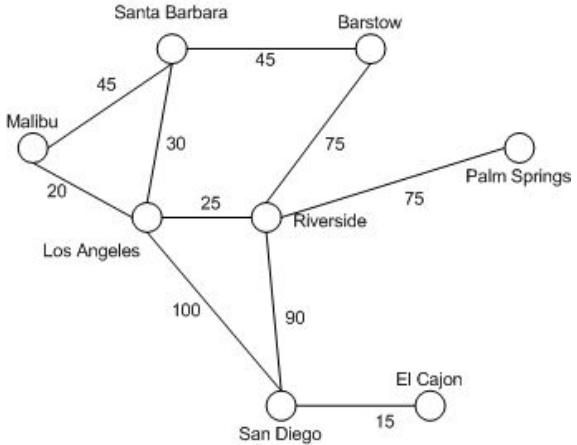
SCC 503 - Alg. Estrut. Dados II

Introdução

- Como podemos representar grafos para resolver problemas práticos?
- Qual a melhor representação?
- Precisamos de uma representação diferente para Grafos e Digrafos ?
- E se o grafo tiver arestas com pesos ?
- Como ler arestas e vértices e armazenar em memória ?
- Vamos começar pela última questão.
    - Como

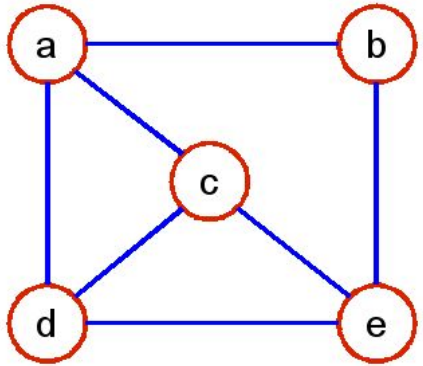






Lendo vértices e arestas de um “arquivo”

Seja o grafo ao lado, representado em um arquivo texto com o seguinte formato:

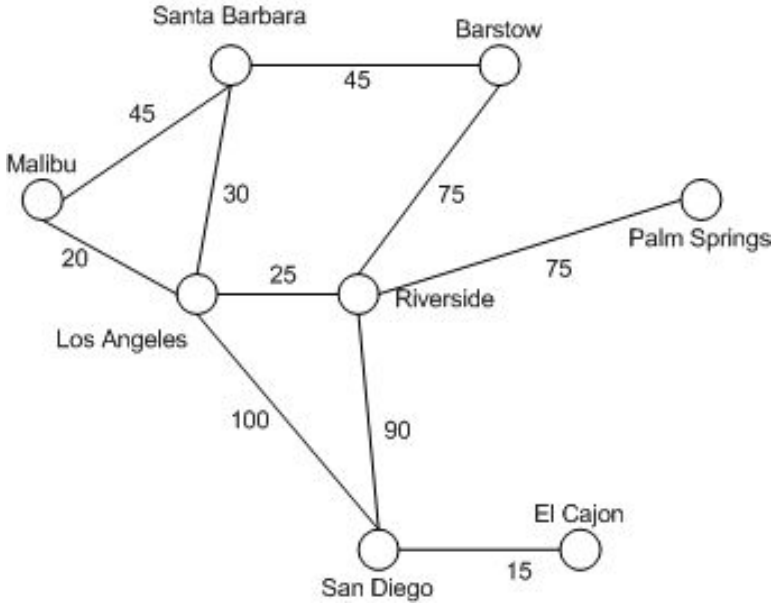



```
5
a b
a c
b e
c d
c e
```

Lendo vértices e arestas de um “arquivo”

Seja o grafo ao lado, representado em um arquivo texto com o seguinte formato:

```
8
0 1 45
0 2 20
1 2 30
1 3 45
2 4 25
2 5 100
3 4 75
4 5 90
4 6 75
5 7 15
```



Lendo vértices e arestas de um “arquivo”

- É bastante improvável que alguém vá querer “digitar” todos vértices e arestas toda vez que for criar um grafo.
- Vamos então escrever um código que leia do arquivo cada um destes valores.
- Como vc faria isso?
  - código exemplo serão deixados na pagina da disciplina

Estrutura de dados

- Será que é preciso de fato diferenciar a estrutura de dados para grafos e digrafos?
  - reveja as definições.
  - Um grafo não direcionado é aquele em que uma resta v-w pode ser vista com um arco não apenas do tipo v-w, como também um arco w-v
- Portanto, as estruturas não precisam ser necessariamente diferentes.
- O que muda é que em um grafo, ao contrário do digrafo, ambos arcos v-w quanto w-v precisarão estar representadas na estrutura.

Estruturas possíveis

- Lista de arcos (ou vértices)
- Lista de adjacências
- Matriz de adjacência
- Matriz de Incidência

Lista de arcos

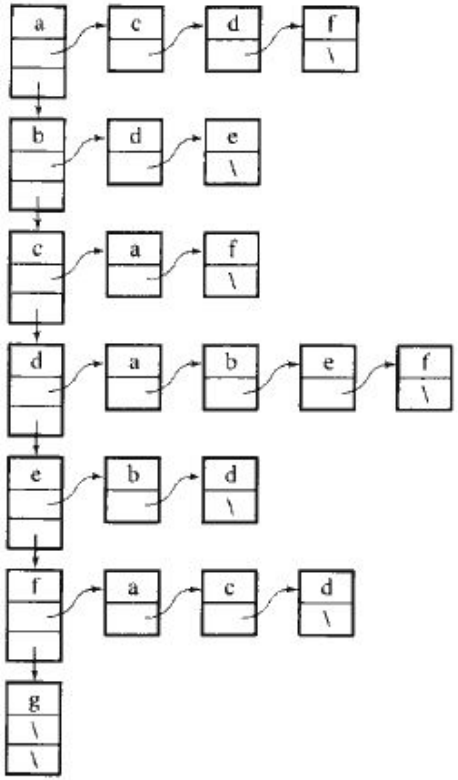
- crie uma lista (pode ser um vetor ou uma estrutura encadeada) com todos os arcos ou arestas existentes no seu (di)grafo.
- A partir daí crie um TAD com funções do tipo:
  - Graph initGraph(int)
  - void destroyGraph(Graph)
  - insertArc(Arc);
  - searchArc(Arc);
  - removeArc(Arc);
  - Graph copyGraph(Graph);
  - etc.…
- Você acha a lista de arcos uma boa opção??
  - e se o grafo possuísse um vértice sem conexão alguma com outro vértice ???

Lista de adjacência

- Especifica os vértices adjacentes a cada vértice do grafo
- Implementações possíveis
  - uma implementação em tabela
  - lista ligada (encadeada) de vértices, como ponteiro para uma lista de adjacências (que também pode ser ligada)
  - um vetor de vértices (estático) com ponteiro para uma lista de adjacências (tabém uma lista ligada)

Lista de adjacência

a	c	d	f	
b	d	e		
c	a	f		
d	a	b	e	f
e	b	d		
f	a	c	d	
g				

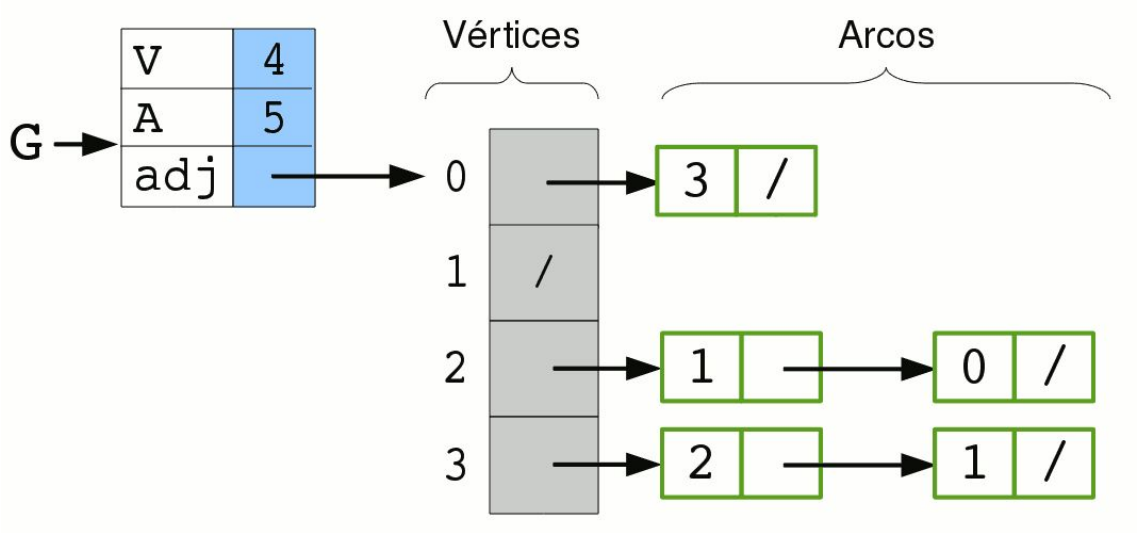


Lista de Adjacência

- Se desejarmos que o vértice e arcos tenham “nomes”, devemos pensar numa estrutura para armazenar isso..
- Estrutura ideal para armazenar grafos esparsos..
  - por que???
  - pense em uma matriz, ao invés de uma estrutura em lista..

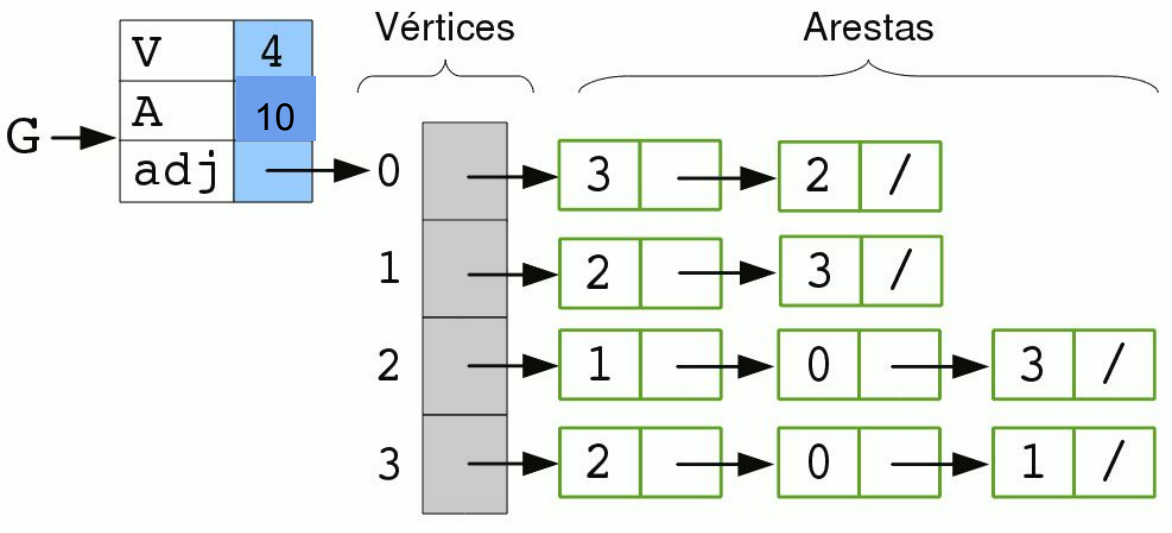
Lista de adjacência para um digrafo

- usamos um vetor de vértices (nao ligada) e a lista de adjacência ligada.
- Voce pode agora completar o seu TAD com funções do tipo
  - initGraph, destroyGraph, insertArc(Arc); searchArc(Arc); removeArc(Arc), etc..
-



Lista de adjacência para um grafo

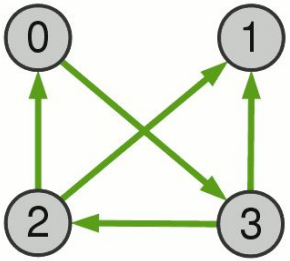
- usamos um vetor de vértices (nao ligada) e a lista de adjacência ligada.
- Voce pode agora completar o seu TAD com funções do tipo
  - initGraph, destroyGraph, insertArc(Arc); searchArc(Arc); removeArc(Arc), etc..
-



Matriz de Adjacência

- matriz binária de tamanho V x V, tal que cada entrada $d_{i,j}$
  - 1, se existe $A(v_i,v_j)$
  - 0. caso contrário
- Pode ser generalizada para multigrafos se ao invés de utilizarmos 0s e 1s indicarmos $d_{i,j}$ = número de arestas entre v_i e v_j
- se o grafo tiver laços, podemos colocar valores na diagonal principal.

Matriz de Adjacência: digrafo

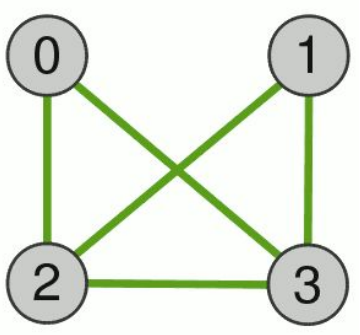


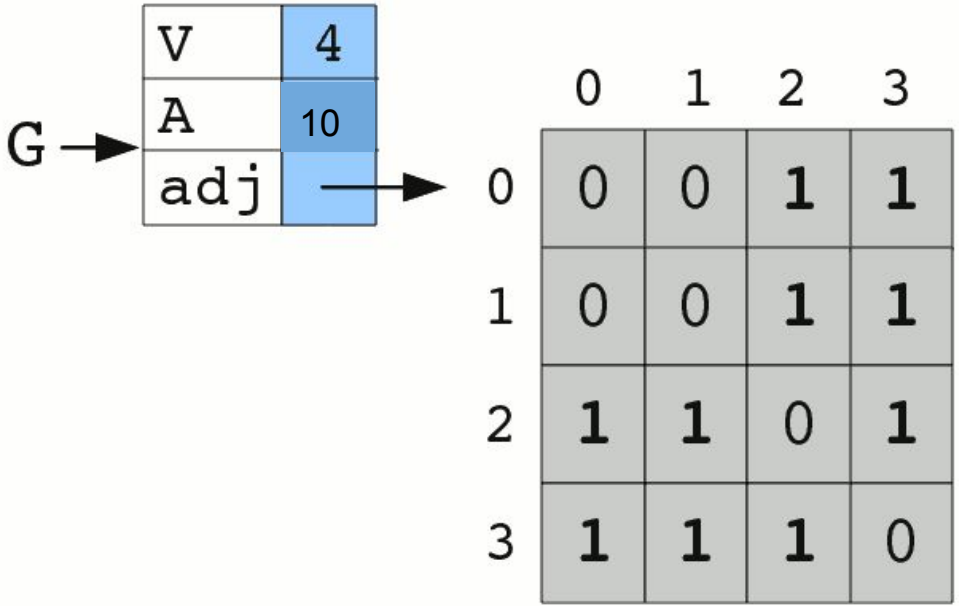
G →

V	4
A	5
adj	→

	0	1	2	3
0	0	0	0	**1**
1	0	0	0	0
2	**1**	**1**	0	0
3	0	**1**	**1**	0

Matriz de Adjacência: grafo

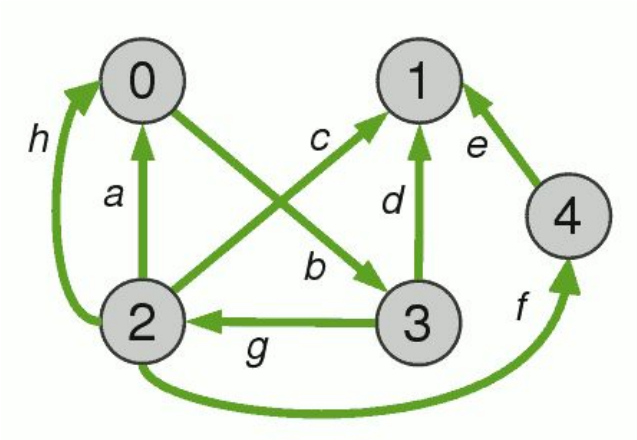




Matriz de Incidência

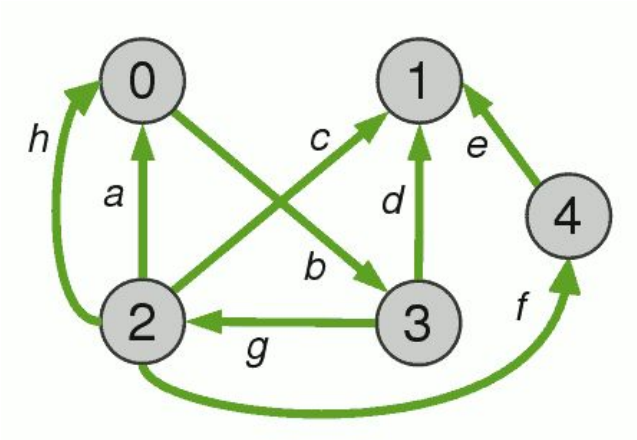
- baseada na incidência de vértices e arestas, é uma matriz de tamanho V × A, tal que cada entrada $c_{i,j}$
  - 1 , se a aresta a_j é incidente com o vértice v_i
  - 0, caso contrário

Matriz de Incidência: digrafo



	a	b	c	d	e	f	g	h
0	1	-1	0	0	0	0	0	1
1	0	0	1	1	1	0	0	0
2	-1	0	-1	0	0	-1	1	-1
3	0	1	0	-1	0	0	-1	0
4	0	0	0	0	-1	1	0	0

Matriz de Incidência: grafo



	a	b	c	d	e	f	g	h
0	**1**	**1**	0	0	0	0	0	**1**
1	0	0	**1**	**1**	**1**	0	0	0
2	**1**	0	**1**	0	0	**1**	**1**	**1**
3	0	**1**	0	**1**	0	0	**1**	0
4	0	0	0	0	**1**	**1**	0	0

Desempenho Assintótico: notação O()

	Lista de Arcos	Lista de Adjacência	Matriz de Adjacência
Espaço	$n + m$	$n + m$	$n^2 + m$
Verificar arcos incidentes	m	grau(v)	n
Verificar adjacência	m	grau(v)	1
Inserir vértice	1	1	n^2
Inserir arco	1	1	1
Remover vértice	m	grau(v)	n^2
Remover arco	1	1	1