Estruturas de Dados Clássicas – Grafos – Parte 2

Prof. Bárbara Quintela

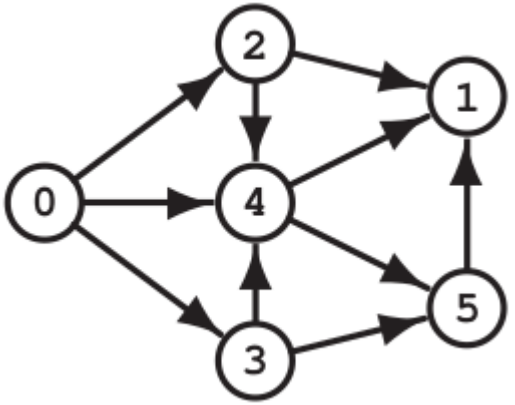
barbaraquintela@pucminas.cesjf.br

Introdução

- Como podemo descobrir algoritmicamente se é possível ir de um vértice a outro em um grafo viajando pelas arestas?

- Exemplo:

  No digrafo ao lado o vértice 0 é alcançável a partir de algum outro?



Introdução

- Como podemo descobrir algoritmicamente se é possível ir de um vértice a outro em um grafo viajando pelas arestas?
- Exemplo: digrafo e vetor com lista de adjacencias

0: 3 4 2

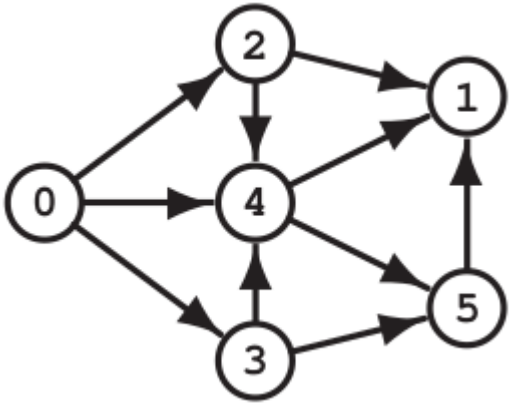
1:

2: 4 1

3: 4 5

4: 1 5

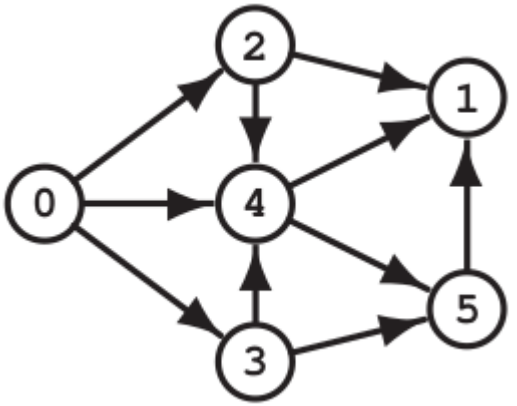
5: 1



Introdução

- Para decidir se vértice 5 está ao alcance de 0 uma possível solução seria chamar uma função que faz uma numeração dos vértices criando rótulos 1 para um vértice que está ao alcance e 0 caso contrário

0 reachR(G,0)	1 - - - - -
0-3 reachR(G,3)	1 - - 1 - -
3-4 reachR(G,4)	1 - - 1 1 -
4-1 reachR(G,1)	1 1 - 1 1 -
4-5 reachR(G,5)	1 1 - 1 1 1
5-1	
3-5	
0-4	
0-2 reachR(G,2)	1 1 1 1 1 1
2-4	
2-1	



* Nesse caso, o estado final da tabela mostra que todos os vértices estão ao alcance do vértice 0.

Busca em Grafos

- Esse exemplo é uma introdução para a ideia de busca em grafos
- Existem basicamente duas técnicas de busca em grafos:
  - Busca em profundidade (depth-first search – DFS)
  - Busca em Largura (breadth-first search - BFS)

- Veremos os dois casos a seguir

Busca em Profundidade

- É um algoritmo de varredura para caminhar em um grafo.
- É utilizada como pré-processamento para resolver diversos problemas.
  - Base para verificação de grafos acíclicos, ordenação topológica e componentes fortemente conectados

Busca em Profundidade

- As arestas são exploradas a partir do vértice v mais recentemente descoberto que ainda possui arestas não exploradas saindo dele
- Quando todas as arestas adjacentes tiverem sido exploradas, a busca anda para trás (backtrack) para explorar vértices que saem do qual v foi descoberto
- O processo continua até que sejam descobertos todos os vértices que sejam alcançáveis a partir do original

Busca em Profundidade

- Um esquema de cores é utilizado para diferenciar vértices durante a busca:
  - Todos são inicializados com branco
  - Quando é descoberto a primeira vez torna-se cinza
  - Quando a lista de adjacentes for completamente examinada vira preto
- Também registra o momento em que foi descoberto
  - d[v] – tempo em que é tornado cinza
  - t[v] – tempo em que a busca termina

Busca em Profundidade

- A variável **tempo** é utilizada para marcar o tempo de descoberta e de término
- Os tempos são utilizados em algoritmos para grafos e são úteis para acompanhar o comportamento da busca

Busca em Profundidade

No procedimento busca em profundidade:

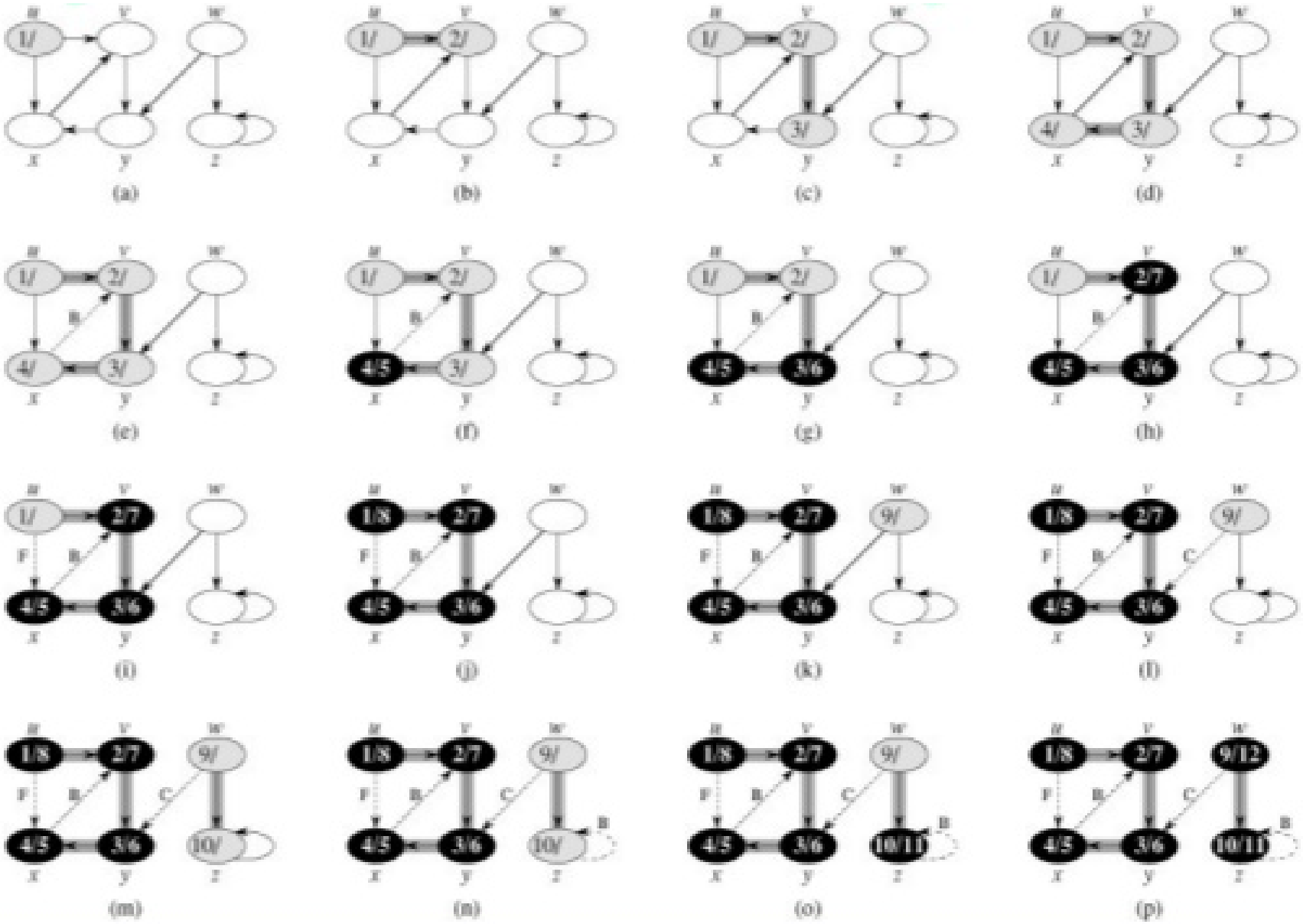
- A variável tempo é usada para registrar os tempos de descoberta e de término é inicializada com zero
- O primeiro laço for colore todos os vértices de branco e inicializa antecessores para -1
- O laço for seguinte verifica cada vértice em V e visita todos os vértices brancos
  - Toda vez que visitaDFS() é chamado o vértice u se torna raiz de uma nova árvore de busca em profundidade

Busca em Profundidade

Em cada chamada de visitaDFS()

- O vertice u é inicialmente branco
- Na primeira linha é tornado cinza
- A variável tempo é incrementada
- Novo valor de tempo é registrado como tempo de descoberta d[u]
- Comando if examina lista de vertices v adjacentes a u e visita recursivamente v se ele for branco

Busca em Profundidade - Exemplo



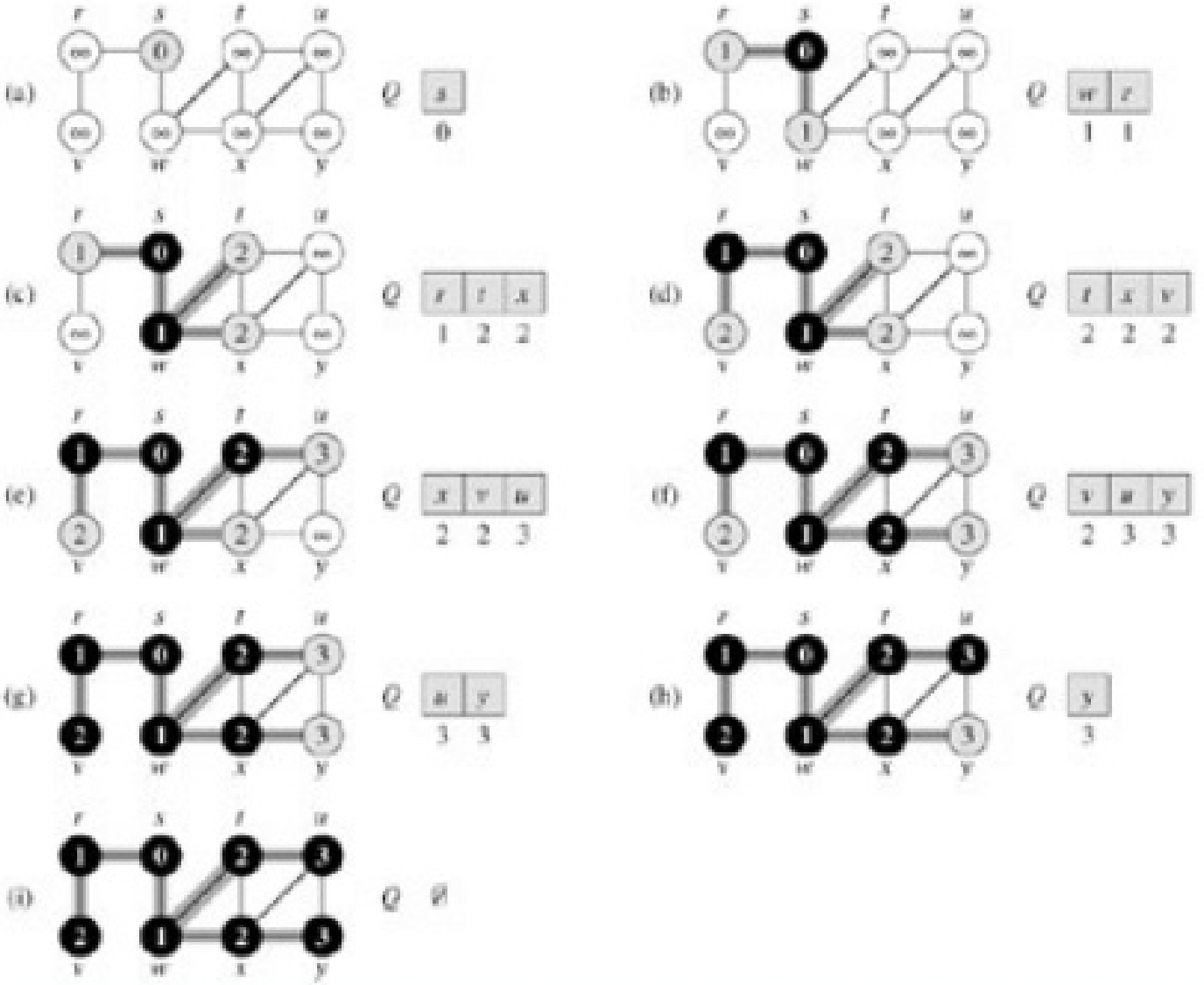
Busca em Largura

- Expande a fronteira entre vértices descobertos e não descobertos uniformemente
- Como se fossem círculos concêntricos
- Base para outros algoritmos importantes como
  - Algoritmo de Prim para obter árvore geradora mínima
  - Algoritmo de Djikstra para obter o caminho mais curto de um vértice a todos os outros
- O grafo pode ser direcionado ou não direcionado

Busca em Largura

- Assim como na busca em profundidade, os vértices iniciam brancos
- Quando um vértice é descoberto pela primeira vez, se torna cinza
- Todos os vertices adjacentes a vertices pretos já foram descobertos
- O algoritmo apresentado usa uma fila para gerenciar os vértices cinza

Busca em Largura



Busca em Largura

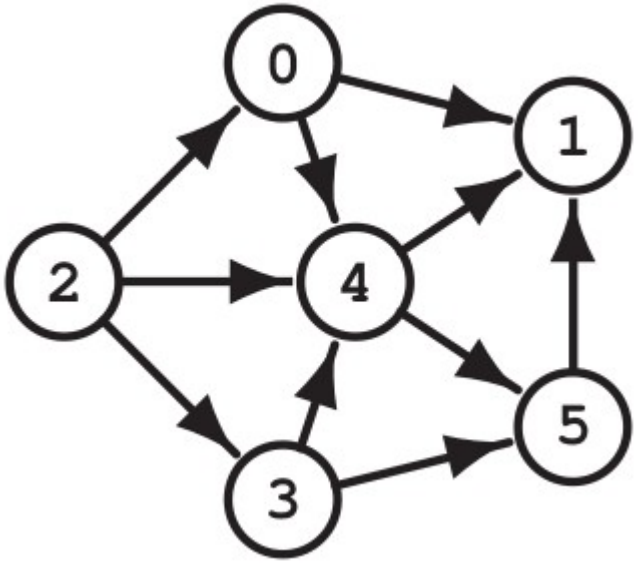
No procedimento busca em largura

- o primeiro *for* colore todos os vértices de branco
- Inicializa a distancia dist[u] do vértice origem até o vértice u para um valor “infinito”
- Inicializa antecessores para -1 (nil)
  - Se u não tem antecessor como nos casos que u corresponde ao vértice origem, ou u ainda não foi descoberto, então ant = -1

Exemplo

Inicialize o grafo dirigido a seguir e confira se a ordem de descoberta da busca em profundidade é como segue:

0 1 4 5 2 3



Exemplos

Inicialize o grafo dirigido a seguir e confira se o lista da busca em largura é como segue:

Fila:

0

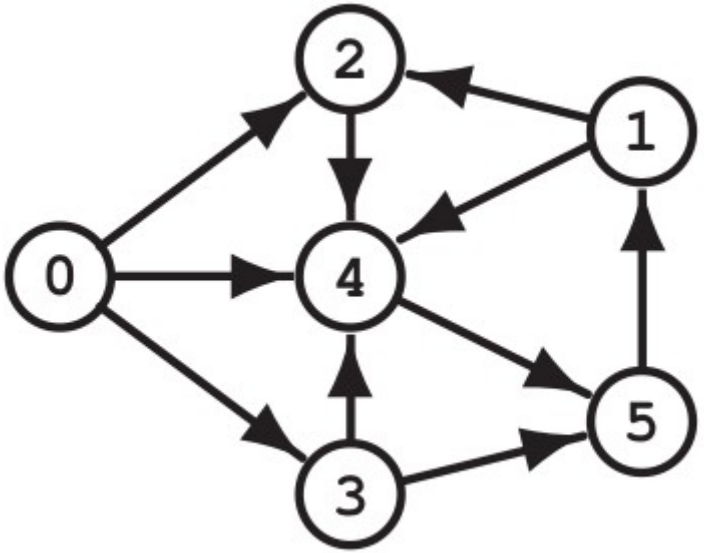
2 3 4

3 4

4 5

5

1



Exercícios

- Implemente as funções de busca em profundidade para grafos armazenados como matriz de adjacência
- Implemente as funções de busca em largura para grafos armazenados como matriz de adjacência.

Estruturas de Dados Clássicas – Grafos – Parte 2

Prof. Bárbara Quintela

barbaraquintela@pucminas.cesjf.br