

GRAFOS – 25/2

Ciência da Computação
Universidade do Vale do Itajaí – UNIVALI

Profª Fernanda dos Santos Cunha
fernanda.cunha@univali.br

1

1

Grafos: Unidade 7 – Busca em Grafos

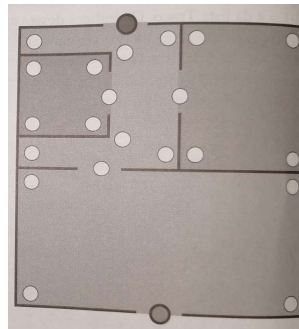
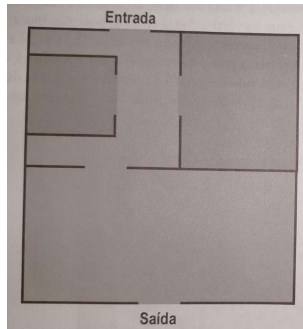
- ▣ Procedimento para examinar com eficiência vértices e arestas de um grafo
- ▣ Algoritmo geral de busca

```
Ler  $G=(V,A)$   
Escolher e marcar um vértice  $i$   
Enquanto existir  $j \in V$  marcado c/uma aresta  $(j,k)$  não explorada Faça  
    Escolher o vértice  $j$  e explorar a aresta  $(j,k)$   
    // condição variável em conformidade com o tipo de busca  
    Se  $k$  é não marcado então marcar  $k$   
Fim_Enquanto
```

2

Busca em Grafos

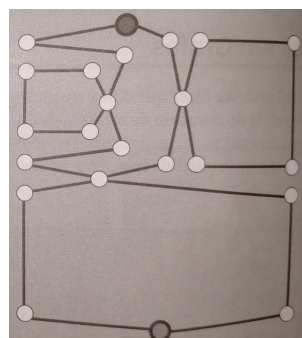
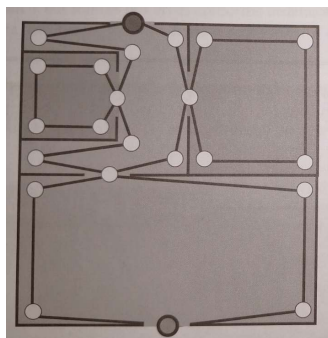
- ❑ Encontrar a saída de um labirinto
- ❑ Considera-se que a circulação seja margeando as paredes. Assim tem-se os pontos de mudança de direção:



3

Busca em Grafos

- ❑ O grafo abaixo representa o labirinto, as arestas mostram movimentos possíveis entre os pto de mudança de direção.

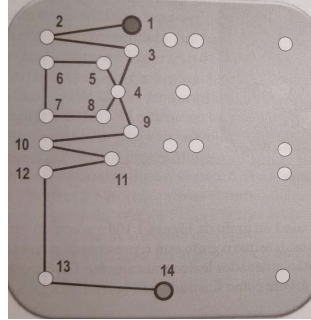
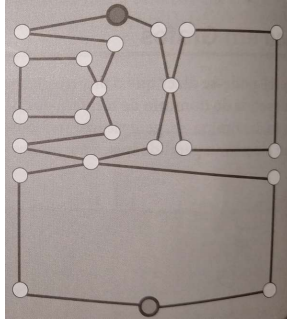


- ❑ A ideia é sair sem nunca percorrer mais de 1 vez a mesma parede = rotular as arestas percorridas sem nunca percorrer uma aresta rotulada.

4

Busca em Grafos

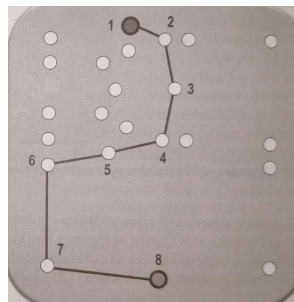
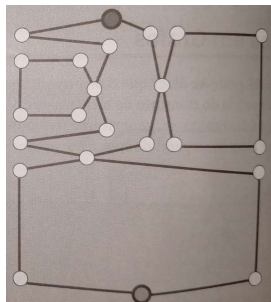
- Aplicando o algoritmo geral tem-se: $(1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (8,4)$ – fecha ciclo, e segue $(4,9), (9,10), (10,11), (11,12), (12,13), (13,14)$.



5

Busca em Grafos

- Uma busca mais eficiente seria:



- De maneira prática, bastaria riscar as paredes de um labirinto sem nunca retirar o lápis da superfície para não se perder – basta nunca riscar duas vezes uma mesma parede.

6

Algoritmos básicos

Dependendo do critério usado para examinar vértices e arestas, diferentes tipos de busca foram desenvolvidos:

- ▣ **BFS Breadth First Search** – busca em largura (amplitude)
 - Camadas horizontais a partir do vértice inicial
- ▣ **DFS Depth First Search** – busca em profundidade
 - Camadas verticais a partir do vértice inicial

7

BFS – Largura

Critério de escolha do nó marcado obedece a: “dentre todos os nós marcados e incidentes a algum arco ainda não explorado, escolhe-se aquele MENOS recentemente alcançado na busca”.
(FILA)

Passo a passo básico:

- Escolhe-se um vértice para o início do caminhamento;
- visitam-se os vértices adjacentes, marcando-os como visitados;
- coloca-se cada um dos vértices adjacentes numa **fila (FIFO)**;
- após visitados os vértices adjacentes, o primeiro da fila torna-se o novo vértice inicial e reinicia o processo;
- termina quando todos os vértices tiverem sido visitados ou o vértice procurado for encontrado.

8

BFS – Largura

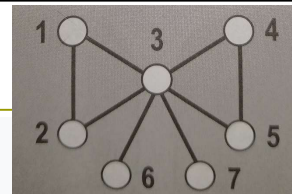
Procedimento BL($G=(N,M)$)

```

definir uma fila Q vazia
escolher o vértice inicial v
marcar v
inserir v em Q
ENQUANTO Q  $\neq \emptyset$  FACA
    v  $\leftarrow$  remove elemento de Q
    PARA todo w  $\in \tau(v)$  FACA
        SE w é não marcado ENTAO
            explorar (v,w)
            inserir w em Q
            marcar w
        SENAO
            SE (v,w) não explorada ENTAO
                explorar (v,w)
            FIMSE
    FIMSE
FIMPARA
FIMENQUANTO
    
```

9

BFS – Largura



```

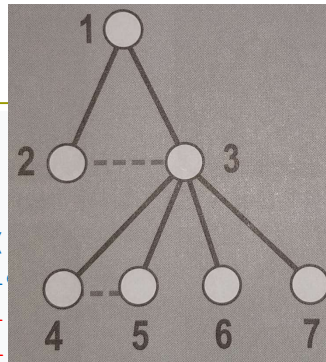
Vértices Marcados= $\emptyset$ ; Fila(Q)= $\emptyset$ 
Marcados=1; Fila(Q)=1.
Marcados=1; Fila(Q)= $\emptyset$ ;  $\tau(1)=2,3$ 
Marcados=1,2; Fila(Q)=2; explora(1,2)
Marcados=1,2,3; Fila(Q)=2,3; explora(1,3)
Marcados=1,2,3; Fila(Q)=3;  $\tau(2)=1,3$ ; explora(2,3)
Marcados=1,2,3; Fila(Q)= $\emptyset$ ;  $\tau(3)=1,2,4,5,6,7$ 
Marcados=1,2,3,4; Fila(Q)=4; explora(3,4)
Marcados=1,2,3,4,5; Fila(Q)=4,5; explora(3,5)
Marcados=1,2,3,4,5,6; Fila(Q)=4,5,6; explora(3,6)
Marcados=1,2,3,4,5,6,7; Fila(Q)=4,5,6,7; explora(3,7)
Marcados=1,2,3,4,5,6,7; Fila(Q)=5,6,7;  $\tau(4)=3,5$ ;
explora(4,5)
Marcados=1,2,3,4,5,6,7; Fila(Q)=6,7;  $\tau(5)=3,4$ 
Marcados=1,2,3,4,5,6,7; Fila(Q)=7;  $\tau(6)=3$ 
Marcados=1,2,3,4,5,6,7; Fila(Q)= $\emptyset$ ;  $\tau(7)=3$ 
    
```

10

BFS – Largura

```

Vértices Marcados= $\emptyset$ ; Fila(Q)= $\emptyset$ 
Marcados=1; Fila(Q)=1.
Marcados=1; Fila(Q)= $\emptyset$ ; T(1)=2,3
Marcados=1,2; Fila(Q)=2; explora(
Marcados=1,2,3; Fila(Q)=2,3; expl
Marcados=1,2,3; Fila(Q)=3; T(2)=1
Marcados=1,2,3; Fila(Q)= $\emptyset$ ; T(3)=1
Marcados=1,2,3,4; Fila(Q)=4; explora(3,4)
Marcados=1,2,3,4,5; Fila(Q)=4,5; explora(3,5)
Marcados=1,2,3,4,5,6; Fila(Q)=4,5,6; explora(3,6)
Marcados=1,2,3,4,5,6,7; Fila(Q)=4,5,6,7; explora(3,7)
Marcados=1,2,3,4,5,6,7; Fila(Q)=5,6,7; T(4)=3,5;
explora(4,5)
Marcados=1,2,3,4,5,6,7; Fila(Q)=6,7; T(5)=3,4
Marcados=1,2,3,4,5,6,7; Fila(Q)=7; T(6)=3
Marcados=1,2,3,4,5,6,7; Fila(Q)= $\emptyset$ ; T(7)=3
    
```



11

BFS – Largura

Olhar as ligações exploradas em azul

```

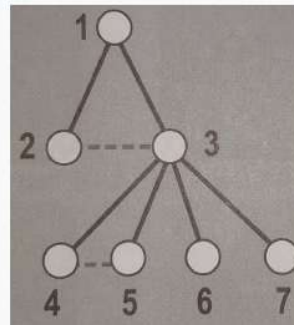
explora(1,2)
explora(1,3)
explora(3,4)
explora(3,5)
explora(3,6)
explora(3,7)
    
```

Mostrar a árvore textualmente

```

1-2
1-3
  3-4
  3-5
  3-6
  3-7
    
```

As ligações em vermelho, que são as arestas de retorno (tracejadas), não aparecem na árvore textual.



12

DFS – Profundidade

Critério de escolha obedece a: “dentre todos os nós marcados e incidentes a algum arco ainda não explorado, escolhe-se aquele MAIS recentemente alcançado”.

Passo a passo básico:

- Escolhe-se um vértice inicial;
- visita-se um primeiro vértice adjacente, marcando-o como visitado;
- coloca-se o vértice adjacente visitado numa **pilha (LIFO)**;
- o vértice visitado torna-se o novo vértice inicial;
- repete-se o processo até que o vértice procurado seja encontrado ou não haja mais vértices adjacentes. Se verdadeiro, desempilha-se o topo e procura-se o próximo adjacente, repetindo o algoritmo;
- o processo termina quando o vértice procurado for encontrado ou quando a pilha estiver vazia e todos os vértices tiverem sido visitados.

13

DFS – Profundidade

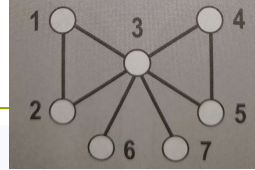
Procedimento BP(v)

```
marcar v
ENQUANTO existir w ∈ T(v) FACA
  SE w é não marcado ENTAO
    explorar (v,w)
    marcar w
    BP(w)
  SENAO
    SE (v,w) não explorada ENTAO
      explorar (v,w)
    FIMSE
  FIME
FIMENQUANTO
```

Quando ocorrem buscas em grafos muito grandes, que não podem ser armazenadas completamente em memória, a busca em profundidade não termina – casos onde o comprimento de um caminho é infinito. Não basta “lembrar quais nós já foram visitados” => ideia é estabelecer um limite de aumento na profundidade da árvore.

14

DFS – Profundidade

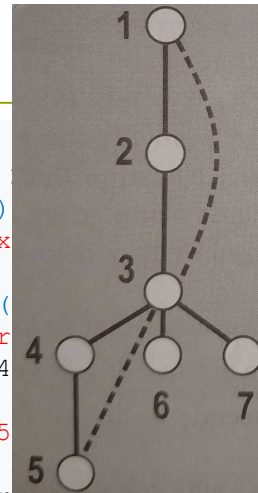


```

Vértices Marcados=∅; BP(1)
Marcados=1; T(1)=2,3; w=2; explora(1,2); BP(2)
Marcados=1,2; T(2)=1,3; w=3; explora(2,3); BP(3)
Marcados=1,2,3; T(3)=1,2,4,5,6,7; w=1; explora(3,1);
w=2; w=4; explora(3,4); BP(4)
Marcados=1,2,3,4; T(4)=3,5; w=5; explora(4,5); BP(5)
Marcados=1,2,3,4,5; T(5)=3,4; w=3; explora(5,3); w=4;
Encerra BP(5); Retoma BP(4); Encerra BP(4); Retoma
BP(3)
Marcados=1,2,3,4,5; T(3)=1,2,4,5,6,7; w=5; w=6;
explora(3,6); BP(6)
Marcados=1,2,3,4,5,6; T(6)=3; w=3; Encerra BP(6);
Retoma BP(3)
Marcados=1,2,3,4,5,6; T(3)=1,2,4,5,6,7; w=7;
explora(3,7); BP(7)
Marcados=1,2,3,4,5,6,7; T(7)=3; w=3; Encerra BP(7);
Encerra BP(3); Encerra BP(2); Retoma BP(1)
Marcados=1,2,3,4,5,6,7; T(1)=2,3; w=3; Encerra BP(1)
    
```

15

DFS – Profundidade



```

Vértices Marcados=∅; BP(1)
Marcados=1; T(1)=2,3; w=2; explora(1,2);
Marcados=1,2; T(2)=1,3; w=3; explora(2,3)
Marcados=1,2,3; T(3)=1,2,4,5,6,7; w=1; ex
w=2; w=4; explora(3,4); BP(4)
Marcados=1,2,3,4; T(4)=3,5; w=5; explora(
Marcados=1,2,3,4,5; T(5)=3,4; w=3; explor
Encerra BP(5); Retoma BP(4); Encerra BP(4)
BP(3)
Marcados=1,2,3,4,5; T(3)=1,2,4,5,6,7; w=5
explora(3,6); BP(6)
Marcados=1,2,3,4,5,6; T(6)=3; w=3; Encerra BP(6),
Retoma BP(3)
Marcados=1,2,3,4,5,6; T(3)=1,2,4,5,6,7; w=7;
explora(3,7); BP(7)
Marcados=1,2,3,4,5,6,7; T(7)=3; w=3; Encerra BP(7);
Encerra BP(3); Encerra BP(2); Retoma BP(1)
Marcados=1,2,3,4,5,6,7; T(1)=2,3; w=3; Encerra BP(1)
    
```

16

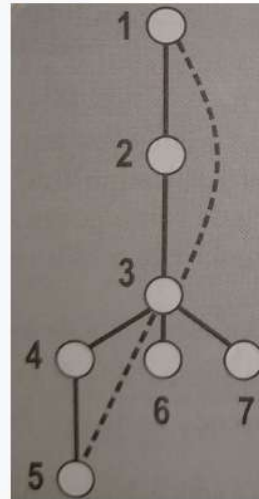
DFS – Profundidade

Olhar as ligações exploradas em azul

```
explora(1,2)
explora(2,3)
explora(3,4)
explora(4,5)
explora(3,6)
explora(3,7)
```

Mostrar a árvore textualmente

```
1-2
  2-3
    3-4
      4-5
    3-6
    3-7
```



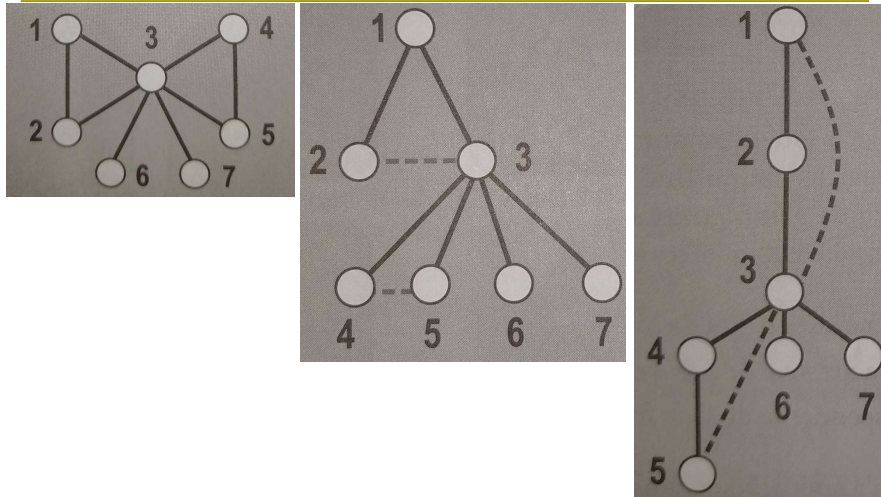
17

BFS versus DFS

- ❑ Apesar da semelhança entre as siglas, as buscas são muito diferentes e têm aplicações muito diferentes.
- ❑ A diferença mais marcante está nas estruturas de dados auxiliares empregadas: a BFS usa uma fila (de vértices), enquanto a DFS usa uma pilha. (Na versão **recursiva** da DFS, a pilha não aparece pois é administrada pelo mecanismo de recursão.)
- ❑ Outras diferenças mais superficiais:
 - na BFS, o usuário escolhe o vértice inicial; na DFS o próprio algoritmo escolhe o vértice inicial de cada etapa;
 - a DFS visita todos os vértices acessíveis a partir do nó inicial em uma determinada direção, para só depois explorar outros caminhos, conforme a necessidade. Enquanto que a BFS garante que todos os vértices que estão a uma distância k do vértice inicial são visitados antes de se procurar os vértices a uma distância $k+1$.
 - em geral, a DFS é descrita em estilo recursivo enquanto a BFS é descrita em estilo iterativo.

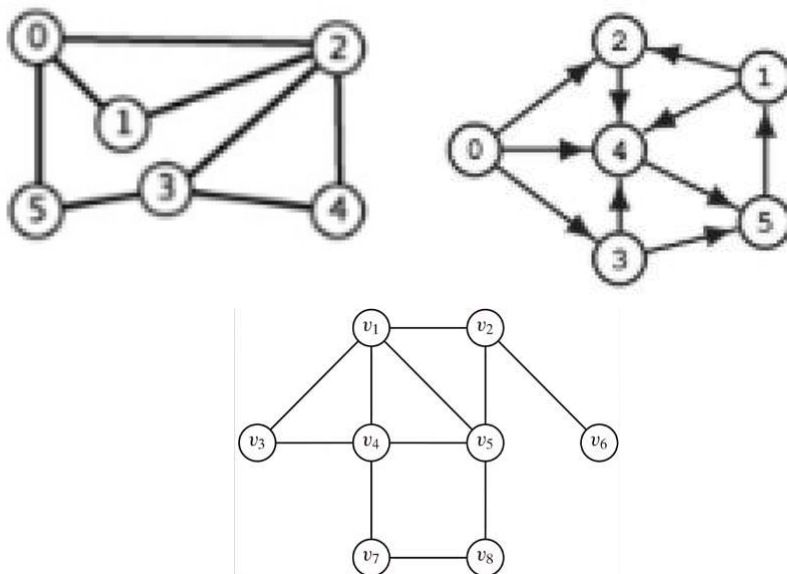
18

BFS versus DFS



19

Exercícios – aplique os algoritmos e mostre as árvores finais



20