

Algoritmos de busca

Seja G um grafo orientado com m vértices, v_1, v_2, \dots, v_m .

Suponha que nos encontramos num vértice v_i do grafo e que queremos encontrar e identificar os vértices que são alcançáveis a partir de v_i .

De seguida, vamos estudar dois algoritmos que permitem atingir este objectivo. São eles o algoritmo de busca em profundidade e o algoritmo de busca em largura.

Algoritmo de busca em largura

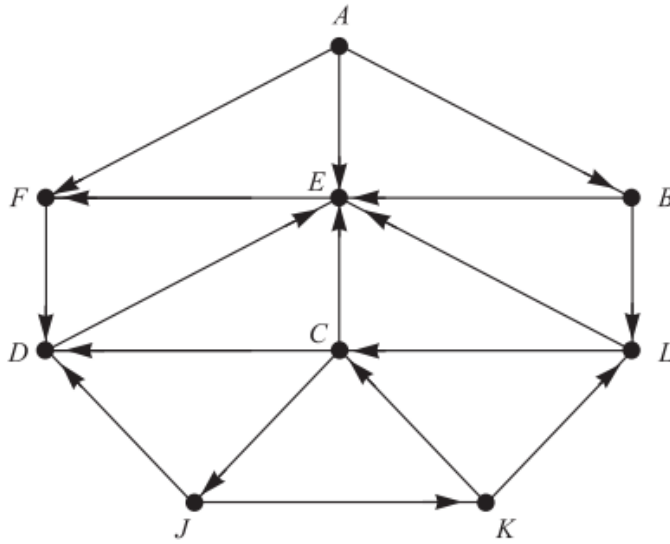
Dado um grafo $G = G(V, E)$ e dado um vértice v_i , a busca em largura começa por visitar primeiro os vértices que são alcançáveis usando apenas uma aresta, de seguida os vértices que são alcançáveis usando duas arestas e assim sucessivamente, até visitar todos os vértices alcançáveis a partir de v_i , ou seja, o algoritmo descobre todos os vértices alcançáveis por um caminho de comprimento k a partir de v_i , antes de descobrir qualquer vértice alcançável por um caminho de comprimento $k + 1$.

Pseudocódigo do algoritmo de busca em largura:

Este algoritmo executa a busca em largura num grafo G a partir de um vértice de partida A .

- 1 - Colocar todos os vértices com estado "não visitado" (símbolo 1);
- 2 - Colocar o vértice de partida A na FILA e alterar o seu estado para estado "em espera" (símbolo 2);
- 3 - Enquanto a FILA não estiver vazia, realizar o seguinte:
 - 3.1 - remover o vértice N da frente da FILA e alterar o seu estado para "processado" (símbolo 3);
 - 3.2 - avaliar os sucessores J de N :
 - 3.2.1 - se $estado(J) = 1$, inserir o vértice J no fim da FILA e fazer $estado(J) = 2$;
 - 3.2.1 - se $estado(J) = 2$ ou $estado(J) = 3$, ignorar o vértice J ;

Exemplo: Considere o grafo G representado na seguinte figura:



Adjacency lists
$A: B, E, F$
$B: E, L$
$C: D, E, J$
$D: E$
$E: F$
$F: D$
$J: D, K$
$K: C, L$
$L: C, E$

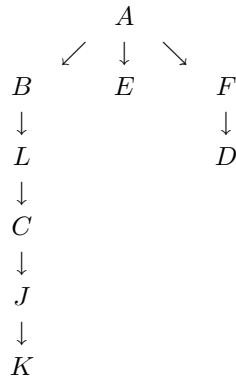
Vamos realizar o algoritmo de busca em largura a partir do vértice A :

	Vértice visitado	FILA(\leftarrow)
iterada 1		A
iterada 2	A	B^A, E^A, F^A
iterada 3	B^A	E^A, F^A, L^B
iterada 4	E^A	F^A, L^B
iterada 5	F^A	L^B, D^F
iterada 6	L^B	D^F, C^L
iterada 7	D^F	C^L
iterada 8	C^L	J^C
iterada 9	J^C	K^J
iterada 10	K^J	\emptyset

Esta busca deu origem à seguinte ordem

$$A - B - E - F - L - D - C - J - K,$$

que pode ser representada pela seguinte árvore:



Algoritmo de busca em profundidade

Dado um grafo $G = G(V, E)$ e dado um vértice v_i , na busca em profundidade visitamos sempre primeiro os vértices adjacentes ao último vértice visitado. Neste caso, o objetivo é explorar o grafo em profundidade.

Pseudocódigo do algoritmo de busca em profundidade:

Este algoritmo executa a busca em profundidade num grafo G a partir de um vértice de partida A .

- 1 - Colocar todos os vértices com estado "não visitado" (símbolo 1);
- 2 - Colocar o vértice de partida A na PILHA e alterar o seu estado para estado "em espera" (símbolo 2);
- 3 - Enquanto PILHA não estiver vazia, realizar o seguinte:
 - 3.1 - remover o vértice N do topo da PILHA e alterar o seu estado para "processado" (símbolo 3);
 - 3.2 - avaliar os sucessores J de N :
 - 3.2.1 - se $estado(J) = 1$, inserir o vértice J no topo da PILHA e fazer $estado(J) = 2$;
 - 3.2.1 - se $estado(J) = 2$, eliminar o velho J e inserir o novo J no topo da PILHA;
 - 3.2.1 - se $estado(J) = 3$, ignorar o vértice J .

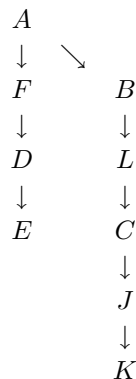
Exemplo: Realizar uma busca em profundidade ao grafo G do exemplo anterior, a partir do vértice A .

	Vértice visitado	PILHA (\leftarrow)
iterada 1		A
iterada 2	A	B^A, E^A, F^A
iterada 3	F^A	B^A, E^A, D^F
iterada 4	D^F	B^A, E^D
iterada 5	E^D	B^A
iterada 6	B^A	L^B
iterada 7	L^B	C^L
iterada 8	C^L	J^C
iterada 9	J^C	K^J
iterada 10	K^J	\emptyset

Esta busca deu origem à seguinte ordem

$$A - F - D - E - B - L - C - J - K,$$

que pode ser representada pela seguinte árvore:



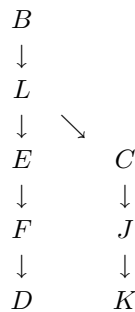
Exemplo: Realizar uma busca em profundidade ao grafo G do exemplo anterior, a partir do vértice B .

	Vértice visitado	PILHA (\leftarrow)
iterada 1		B
iterada 2	B	E^B, L^B
iterada 3	L^B	C^L, E^L
iterada 4	E^L	C^L, F^E
iterada 5	F^E	C^L, D^F
iterada 6	D^F	C^L
iterada 7	C^L	J^C
iterada 8	J^C	K^J
iterada 9	K^J	\emptyset

Esta busca deu origem à seguinte ordem

$$B - L - E - F - D - C - J - K,$$

que pode ser representada pela seguinte árvore:



Exercícios propostos

1. Construa uma função com o nome **blargura** que execute o algoritmo de busca em largura, tendo por base a matriz A de adjacências de um grafo orientado G , ou seja, ao executar o comando **blargura(A,vert)**, deverá ser fornecida uma lista com os vértices alcançáveis partindo do vértice **vert**, segundo a ordem pela qual foram visitados.
 Aplique a função ao exemplo dado anteriormente, sabendo que a respectiva matriz de adjacências se encontra no ficheiro "grafo_f8.xls".
2. Construa uma função com o nome **bprofundidade** que execute o algoritmo de busca em profundidade, tendo por base a matriz A de adjacências de um grafo orientado G , ou seja, ao executar o comando **bprofundidade(A,vert)**, deverá ser fornecida uma lista com os vértices alcançáveis partindo do vértice **vert**, segundo a ordem pela qual foram visitados.
 Aplique a função ao exemplo dado anteriormente, sabendo que a respectiva matriz de adjacências se encontra no ficheiro "grafo_f8.xls".