

SCC-203 ALGORITMOS E ESTRUTURAS DE DADOS II

Prova 1 - Gabarito

Nome: _____ Nro. USP _____

1) O matemático húngaro Paul Erdős (1913-1996), um dos mais brilhantes do século XX, é considerado o mais prolífico da história. Erdős publicou mais de 1.500 artigos, em colaboração com cerca de outros 450 matemáticos. Em sua homenagem, os matemáticos criaram o "número de Erdős". Todo matemático que publicou um artigo com Erdős tem número de Erdős 1. Os que não possuem número 1, mas escreveram um artigo com alguém que possui número 1, possuem número 2, e assim por diante. Quando nenhuma ligação pode ser estabelecida entre Erdős e um matemático, diz-se que este possui número de Erdős infinito. Por exemplo, o número de Erdős de Albert Einstein é 2; e, talvez surpreendentemente, o número de Erdős de Bill Gates é 4.

Considere um subconjunto de 12 matemáticos distintos identificados por A, B, ..., K e L. A lista abaixo informa os que têm artigos em comum:

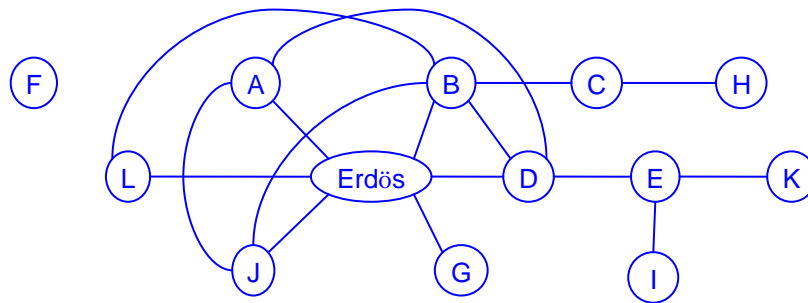
- o autor A tem artigos com D e J;
- o autor B tem artigos com C, D, J e L;
- o autor C tem artigos com H;
- o autor D também tem artigos com E;
- o autor E também tem artigos com I e K.

Erdős tem artigos com os autores A, B, D, G, J e L (e, logicamente, cada um desses autores tem artigo com Erdős; idem para os demais casos).

Considere o problema de determinar o número de Erdős de um autor. Faça:

- (a) (1.0) Modele a situação dos autores acima por meio de um grafo, indicando o que representam os vértices e as arestas (desenhe o grafo completo).

Para que o grafo sirva à pergunta feita, o grafo seria não direcionado, com autores como vértices e arestas entre autores que publicaram juntos, incluindo um vértice para Erdős.



- (b) (1.0) Explique qual conceito/ algoritmo de teoria dos grafos pode ser usado e como pode ser usado para resolver o problema.

O problema de descobrir o número de Erdős de um dado autor é equivalente a determinar o comprimento do caminho mínimo entre 2 vértices no grafo (no caso, os 2 vértices representam 2 autores, Erdős e o outro). Nesse caso, cada aresta percorrida em direção a um autor a partir de Erdős conta um a mais no número de Erdős do autor procurado. O algoritmo de Dijkstra ou outro algoritmo de caminho mínimo pode ser usado para resolver o problema. Opcionalmente, pode-se fazer uma busca em largura, do matemático em questão, a partir do vértice de Erdős, computando a distância percorrida.

(c) (0.5) Com base na especificação da questão, no seu modelo no item (a) e em sua argumentação no item (b), responda: qual o número de Erdős dos matemáticos C e F? Justifique sua resposta.

Analisando o grafo, observa-se que o autor C tem número de Erdős igual a 2, pois há um caminho mínimo de 2 arestas entre C e Erdős. Como F é um componente desconectado do grafo e não há um caminho mínimo entre Erdős e F, diz-se que F tem número de Erdős infinito.

2) O Algoritmo abaixo faz percurso em profundidade, pelos vértices de um grafo não direcionado dado, a partir de um vértice v.

```
Dado G(V,A):
Prof(v)
    marcar v;
    para cada w ∈ ListaAdjacencia(v) faça
        se w é não marcado então
            Prof (w);
    fim /*Prof(v)*/
```

Como o algoritmo acima poderia ser usado/modificado para:

(a) (0.5) Verificar se o grafo G é cíclico ou acíclico. Justifique.

Resp.:

Por ser um grafo não-direcionado, não basta apenas verificar que um vértice adjacente, w, já está marcado para que o grafo tenha um ciclo. É preciso saber se o adjacente já marcado ainda está sob análise (ainda é cinza), pois nesse caso, não representa um ciclo, já que se chegou a v pela aresta (w,v). Assim, é preciso saber se v não foi o último vértice para quem se chamou Prof() – o que implica em ciclo. Isso poderia ser feito por meio de um novo parâmetro na função, que traria o antecessor no caminho, ou por meio de uma pilha explícita, conforme dado em aula.

Considerando o uso de uma pilha P:

Prof(v)

```
marcar v; empilhar v em P;
para cada w ∈ ListaAdjacencia(v) faça
    se w é não marcado então
        Prof (w);
    senão se w não antecede v na pilha P
        então há um ciclo em G
```

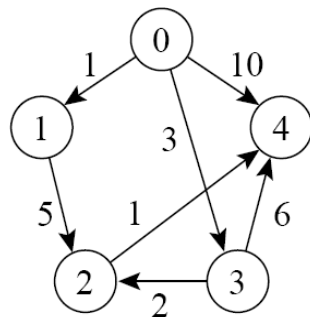
fim /*Prof(v)*/

(b) (0.5) Verificar se o grafo G é conexo ou não.

Resp.:

Ao chamar Prof(v) para um vértice v inicial, se, no final, ele não tiver passado por todos os vértices do grafo, então ele não é conexo. Ao chamar para os vértices restantes, até que todos tenham sido visitados, obtêm-se as componentes conexas do grafo

3) Considere o dígrafo valorado abaixo:



Faça:

(a) (1.5) Calcule os custos dos caminhos mínimos, e as sequências dos mesmos, para o grafo a partir do vértice 0 aplicando o algoritmo de Dijkstra. Preencha os arrays abaixo:

S(sequência):	0 1 2 3 4
	0 1 3 2 4
d(custo mínimo):	0 1 5 3 6
	0 1 5 3 6
Antecessor:	-1 0 3 0 2
	-1 0 3 0 2

Caminho Mínimo de 0 a 1: <0,1>
 Caminho Mínimo de 0 a 2: <0,3,2>
 Caminho Mínimo de 0 a 3: <0,3>
 Caminho Mínimo de 0 a 4: <0,3,2,4>

(b) (1.0) Aplique o algoritmo de ordenação topológica no grafo. Considere as listas de adjacência ordenadas crescentemente pelos números dos vértices, e use uma pilha para armazenar os vértices com grau de entrada igual a zero.

Ordenação Topológica Resultante: 0, 3, 1, 2, 4

(c) Considerando a implementação de dígrafo com matriz de adjacência, faça:

i) (0.5) Mostre a Matriz Adjacência, X, do dígrafo

```
0 1 0 3 10
0 0 5 0 0
0 0 0 0 1
0 0 2 0 6
0 0 0 0 0
```

ii) (0.5) Mostre a Matriz Caminho, P, do dígrafo, conforme é calculada pelo Algoritmo de Roy-Warshall (RW).

```
0 1 1 1 1
0 0 1 0 1
0 0 0 0 1
0 0 1 0 1
```

0 0 0 0 0

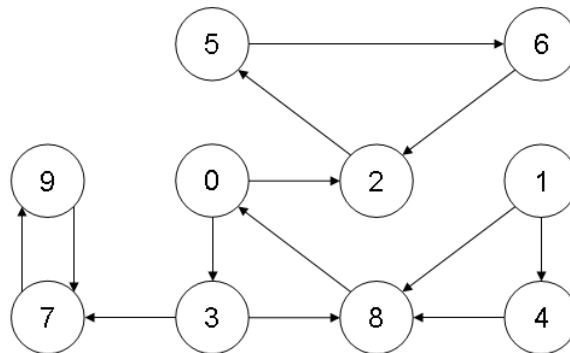
iii) (1.0) Mostre a Matriz dos Custos dos Caminhos Mínimos, MC, do dígrafo, conforme é calculada pelo Algoritmo de Floyd-Warshall (FW).

∞ 1 5 3 6
 ∞ ∞ 5 ∞ 6
 ∞ ∞ ∞ ∞ 1
 ∞ ∞ 2 ∞ 3
 ∞ ∞ ∞ ∞ ∞

iv) (1.0) Mostre a Matriz de Especificação dos Caminhos Mínimos, M, do dígrafo, conforme é calculada pelo Algoritmo FW⁺. Use -1 no lugar de 0, para não confundir com o vértice 0.

-1 1 3 3 3
-1 -1 2 -1 2
-1 -1 -1 -1 4
-1 -1 2 -1 2
-1 -1 -1 -1 -1

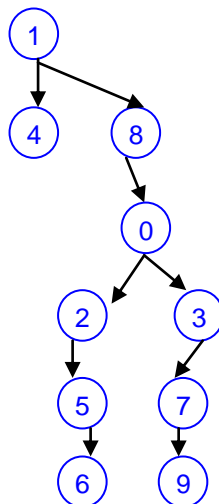
4) Considere o grafo abaixo. Considere também que as listas de adjacência estão ordenadas pelos rótulos dos vértices.



Faça/responda:

(a) (0.5) Monte a(s) árvore(s) de busca em largura quando a busca é feita a partir do vértice 1, e exiba a sequência dos vértices visitados.

Sequência: 1,4,8,0,2,3,5,7,6,9



(b) (0.5) Monte a(s) árvore(s) de busca em profundidade quando a busca é feita a partir do vértice 1, e exiba a sequência dos vértices visitados.

Sequência: 1,4,8,0,2,5,6,3,7,9

