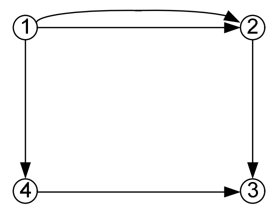
INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Serra

Ministério
da Educação

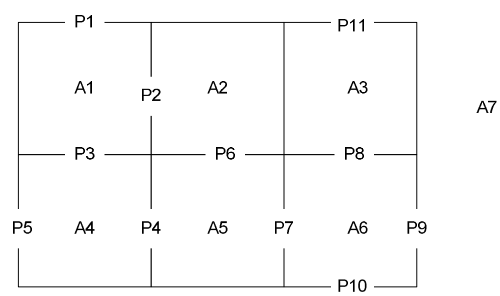
Matemática Discreta - BSI
Prova 4 – Antigas

1) Para o grafo abaixo, seja A a matriz e adjacências:



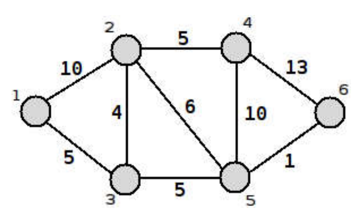
a) Encontre A^2.
b) Encontre $A^{(2)}$.
c) Diga o que significa o elemento da linha 1 e coluna 3 na matriz A^2.

2) Abaixo temos ilustrado a planta baixa de uma casa, com as portas (P1, ... , P11) que áreas (A1, ... , A7).



a) Faça um grafo para representar a planta da casa, onde as áreas são os nós e as portas são as arestas.
b) É possível entrar e sair de cada área da casa de modo que cada porta seja usada exatamente uma vez? Se sim, indique a seguência de portas a ser usada. Se não, justifique. Este problema corresponde ao problema do Ciclo Euleriano ou Ciclo Hamiltoniano?
c) É possível passar por cada área da casa exatamente uma vez e voltar para área inicial? Se sim, indique a seguência de portas a ser usada. Se não, justifique. Este problema corresponde ao problema do Ciclo Euleriano ou Ciclo Hamiltoniano.

3) Considere o grafo abaixo:

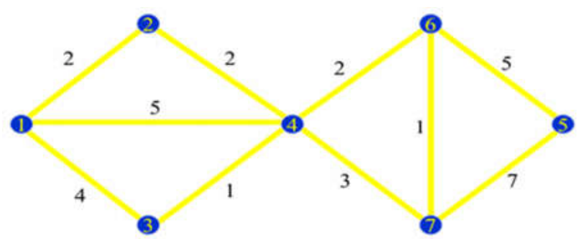


a) Indique a ordem de inserção das arestas na árvore geradora mínima utilizando o Algoritmo de Prim, iniciando no nó 4.
b) Indique a ordem de inserção das arestas na árvore geradora mínima utilizando o Algoritmo de Kruskal.

4) O pseudo-código a seguir representa o Algoritmo de Dijkstra, onde: *n* é o número de nós do grafo *G*, *S* são os vértices para os quais se conhece o caminho mínimo e *R* são os vértices para os quais ainda não se conhece o caminho mínimo;

1. Ler $G = (N,A)$, c_{ij} *(não negativo)* a matriz de custos entre cada para de nós *i* e *j*.
2. Iniciar variáveis $S := \emptyset$; $R := N$;
3.   s = 1; $d(i) = \infty\ \forall i \in N$; $d(s) = 0$ e $pred(s) = 0$;
4. **Enquanto** $|S| < n$ **fazer**
5.   Seja *i* tal que $d(i) = min\{d(j): j \in R\}$;
6. $S := S \cup \{i\}$; $R := R - \{i\}$;
7.   **Para** $j \in N$ *com* $(i,j) \in A$ **fazer**
8.   **Se** $d(j) > d(i)+c_{ij}$ **então**
9. $d(j) = d(i)+c_{ij}$;
10. $pred(j) := i$;
11.   **Fim_Se**
12.   **Fim_Para**
13. **Fim_Enquanto**

Para o grafo abaixo diga o(s) valor(es) de cada variável algoritmo após o término das iterações, iniciando do nó 1.



5) Considerando a busca em grafo, dê a ordem de inspeção dos nós através da busca em profundidade e da busca em nível, do grafo a seguir.

INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Serra

Ministério
da Educação

Caso seja necessário:

Prim:

1. **Ler** $G = (N,A)$, d_{ij} é a "distância" entre os nós vizinhos.
2. Escolha qualquer vértice i.
3. **Iniciar variáveis** $T \leftarrow i$; $V \leftarrow N - \{i\}$;
4. **Enquanto** $T \neq N$ **Para todo** $i \in T$ **fazer**
5. **Encontrar a menor aresta** $(i,k) \in A$ com $i \in T$ e $k \in V$.
6. $T \leftarrow T + \{k\}$
7. $V \leftarrow V - \{k\}$
8. $S \leftarrow S + \{(i,k)\}$
9. **Fim_Enquanto**
10. **Escrever** ***S***

Krukal:

1. **Ler** $G = (N,A)$, d_{ij} é a "distância" entre os nós vizinhos.
2. Ordene os enlaces em ordem não-crescente de distância (d_{ij}) no vertor $H = (h_i)$.
3. **Iniciar variáveis** $T \leftarrow h_1$; i$\leftarrow 2$;
4. **Enquanto** $|T| < n$ **Tome** $h_i \in H$ **fazer**
5. Se $T + h_i$ é um grafo acíclico (árvore) então
6. $T \leftarrow T + h_i$
7. $i \leftarrow i+1$
8. *Caso Contrário*
9. $i \leftarrow i+1$
10. **Fim_Enquanto**
11. **Escrever** T

6)
Em um dado grafo aplicou-se o Algoritmo de busca em Nivel e busca em profundidade, obtendo-se as seguintes sequências.

Profundidade: a, b, c, d, e, f, g, h
Nível: a, b, f, c, d, g, e, h

Sabendo que o grafo em questão é uma árvore binária, apresente a lista de adjacências desta árvore.

7) Considere o tabuleiro 3x4. Com cada quadrado contendo um número

0	4	3	6
7	8	6	8
2	3	1	8

O objetivo do jogo consiste em deslocar uma peça do canto superior esquerdo até o canto inferior direito, através de uma sequencia de movimento para a direita ou para baixo, de forma a minimizar a soma dos pontos adquiridos nos quadrados em que passou.

a) Faço um desenho do grafo que representado este jogo, de forma a solução do jogo seja obtida por um algoritmo de caminho mínimo.
b) Resolva o jogo usando o Algoritmo de Dijkstra

Caso seja necessário, abaixo temos o Algoritmo de Dijkstra.
Algoritmo de Dijkstra

14. Ler $G = (N,A)$, c_{ij} *(não negativo)* é a "distância" entre os nós e nó de origem s.
15. Iniciar variáveis $S := \emptyset$; $R := N$;
16. $d(i) := \infty\ \forall i \in N$; $d(s) := 0$ e $pred(s) := 0$;
17. **Enquanto** $|S| < n$ **fazer**
18. Seja $i \in N$ tal que $d(i) = min\{d(j) : j \in R\}$;

19. $S:=S \cup \{i\}$; $R:=R - \{i\}$;
20. **Para** $j \in N$ *com* $(i,j) \in A$ **fazer**
21. **Se** $d(j) > d(i)+c_{ij}$ **então**
22. $d(j) = d(i)+c_{ij}$;
23. *pred(j):=i;*
24. **Fim_Se**
25. **Fim_Para**
26. **Fim_Enquanto**

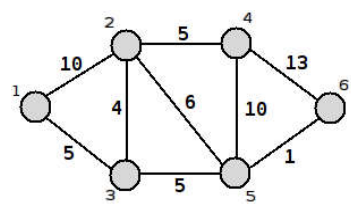
8) Seja **G** um Grafo direcionado e seja **A** sua matriz de adjacências. Prove que o elemento *i,j* da matriz $\mathbf{A}^2$ é igual ao número de caminhos de comprimento 2 do nó *i* para o nó *j*.

9) Um dos algoritmos para encontrar uma árvore geradora mínima é o Algoritmo de Prim.

Algoritmo de Prim

1. **Ler** $G = (N,A)$, d_{ij} é a "distância" entre os nós vizinhos.
2. Escolha qualquer vértice *i*.
3. **Iniciar variáveis** $T \leftarrow i$; $V \leftarrow N - \{i\}$;
4. **Enquanto** $T \neq N$ **Para todo** $i \in T$ **fazer**
5. **Encontrar a menor aresta** $(i,k) \in A$ com $i \in T$ e $k \in V$.
6. $T \leftarrow T + \{k\}$
7. $V \leftarrow V - \{k\}$
8. $S \leftarrow S + \{(i,k)\}$
9. **Fim_Enquanto**
10. **Escrever** ***S*** (arestas da árvore geradora mínima)

a) Diga o que é uma árvore geradora mínima.
b) Explique por que o algoritmo de Prim funcional. Isto é, por que retorna a árvore geradora mínima de um grafo.
c) Faça um "teste de mesa", apresentando o valor das variáveis a cada iteração, para o grafo abaixo.



Caso seja necessário, uso os algoritmos

Algoritmo *EmProfundidade* (grafo *G*, nó *a*)
marque *a* como tendo sido visitado
escreva(*a*)
para cada nó *n* adjacente a *a* **faça**
se nós *n* não tiver sido visitado **então**
EmProfundidade(*G,n*)
fim do *se*
fim do *para*
fim de *Emprofundidade*

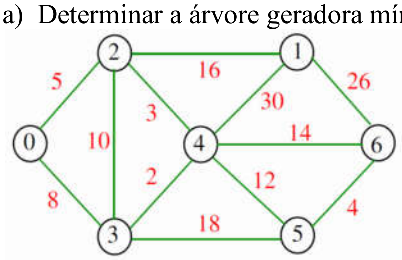
Algoritmo *EmNivel* (grafo *G*, nó *a*)
Variáveis locais: fila de nós *F*
Inicialize *F* como sendo vazio
marque *a* como tendo sido visitado
escreva(*a*); *Insira*(*a,F*)
Enquanto F não é vazio **faça**
para cada nó n adjacente a *frente*(F) **faça**
se nós *n* não foi visitado **então**
marque n como tendo sido visitado
escreva(*a*);*Insira*(*a,F*)
fim do se
fim do para
retire (*F*)
fim do enquanto
fim de EmNivel

10) Considere o seguinte algoritmo:

Algoritmo de Kruskal

1. **Ler** $G = (N,A)$, d_{ij} é a “distância” entre os nós vizinhos.
2. Ordene os enlaces em ordem não-crescente de distância (d_{ij}) no vetor $H = (h_i)$.
3. **Iniciar variáveis** $T \leftarrow h_1$; i $\leftarrow 2;$
4. **Enquanto** $|T| < n$ **Tome** $h_i \in H$ **fazer**
5. Se $T + h_i$ é um grafo acíclico (árvore) então
6. $T \leftarrow T + h_i$
7. $i \leftarrow i+1$
8. *Caso Contrário*
9. $i \leftarrow i+1$
10. **Fim_Enquanto**
11. **Escrever** T (arestas da árvore geradora mínima)

a) Determinar a árvore geradora mínima do seguinte grafo.



b) Apresente uma definição para "árvore geradora máxima" e diga qual deve ser a alteração no algoritmo de Kruskal para obter tal árvore.

11) Considere o algoritmo para percorrer o grafo em profundidade:

Algoritmo *EmProfundidade* (grafo *G*, nó *a*)
marque *a* como tendo sido visitado
escreva(*a*)
para cada nó *n* adjacente a *a* **faça**
    se nós *n* não tiver sido visitado **então**
        EmProfundidade(*G,n*)
    fim do *se*
fim do *para*
fim de *Emprofundidade*

Explique como você faria para usar esse algoritmo para obter uma árvore geradora.

12) É de conhecimento de todos que o Algoritmo de Dijkstra não funciona para arestas com custo negativo. Usando um exemplo, **mostre** e **explique** o porquê dessa afirmação.

Algoritmo de Dijkstra

Ler $G = (N,A)$, c_{ij} *(não negativo)* é a "distância" entre os nós e nó de origem s.
Iniciar variáveis $S := \emptyset$; $R := N$;
    $d(i) := \infty\ \forall i \in N$; $d(s) := 0$ e $pred(s) := 0$;
Enquanto $|S| < n$ **fazer**
    Seja $i \in N$ tal que $d(i) = min\{d(j): j \in R\}$;
    $S := S \cup \{i\}$; $R := R - \{i\}$;
    Para $j \in N$ *com* $(i,j) \in A$ **fazer**
        Se $d(j) > d(i)+c_{ij}$ **então**
            $d(j) = d(i)+c_{ij}$;
            $pred(j) := i$;
        Fim_Se
    Fim_Para
Fim_Enquanto

13)

a) Construa a árvore de Huffman para os caracteres a seguir com as ocorrências dadas e mostre a codificação dos caracteres segundo o código de Huffman:

Caractere	c	d	g	m	z	r
Ocorrência	8000	17500	13500	1000	7500	2500

b) A codificação de Huffman trata de um código de comprimento variável. Para um alfabeto com exatamente esse caracteres, calcule a diferença do número de bits (se houve) entre o código de Huffman e a codificação pela tabela ASCII.

14) Considere a seguinte imagem de 3 bits:

1	1	5	2	5	2	5	1	1	5
6	7	7	0	5	7	7	2	7	1
5	2	5	3	7	1	4	5	3	5
7	7	7	2	1	6	1	5	3	1
5	5	7	1	7	2	1	2	5	5
1	0	1	2	5	1	3	1	0	7
0	5	7	5	0	7	1	3	0	5
5	5	4	0	7	1	5	5	4	4
0	5	1	3	4	2	7	1	2	7
5	7	5	1	0	1	7	1	0	5

a) Obtenha o código de Huffman associado.
b) Calcule o número de bits gastos nessa imagem, para uma codificação fixa de 3 bits.
c) Calcule o número de bits gastos nessa imagem, usando o código de Huffman.