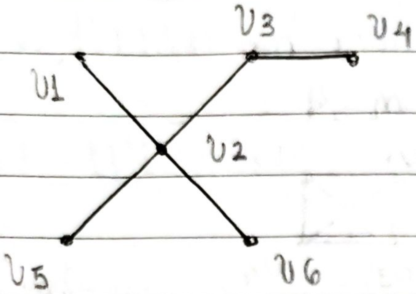
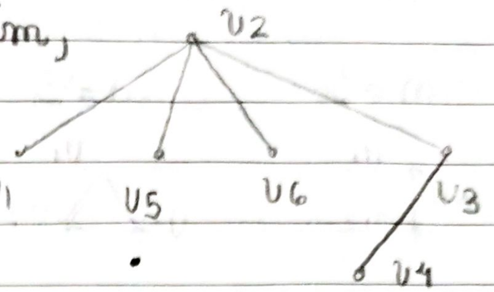
Lista 10

1) Quais dos grafos a seguir são árvores?

a)

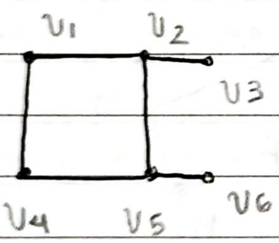


sim,



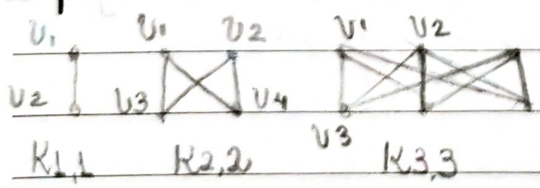
para todo par de vértice existe um único caminho
raiz: v2

b)

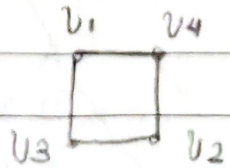


Não é uma árvore, existem dois caminhos possíveis entre qualquer par de vértice, pois há um ciclo
v1 - v2 - v5 - v4.

2) Para quais valores de m e n, o grafo completo bipartido, de m e n vértices é uma árvore?



Em $K_{1,1}$ existe um único caminho entre v_1 e v_2.
Em $K_{2,2}$ representado como



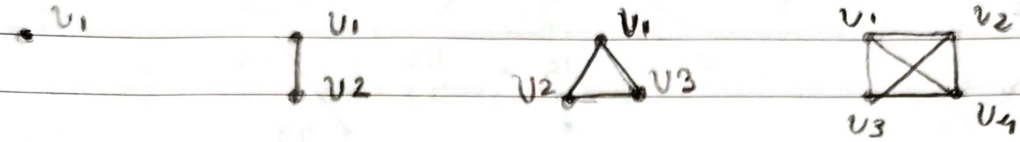
, existe um ciclo, v1 - v4 - v2 - v3, logo, não é árvore. Note que a partir daí todo grafo bipartido completo também possui esse ciclo, então apenas para $m = n = 1$, é árvore.

tilibra

3) Para quais valores de n o grafo completo de n vértices é uma árvore?

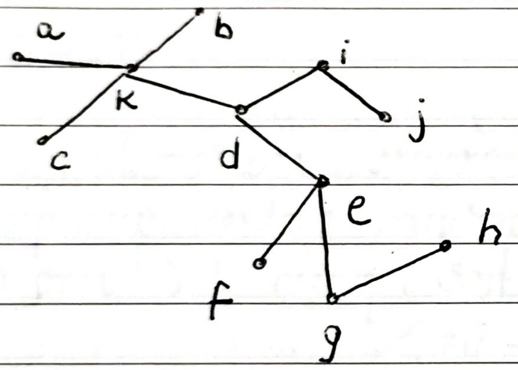
K_n é uma árvore para $n < 3$, há árvore.

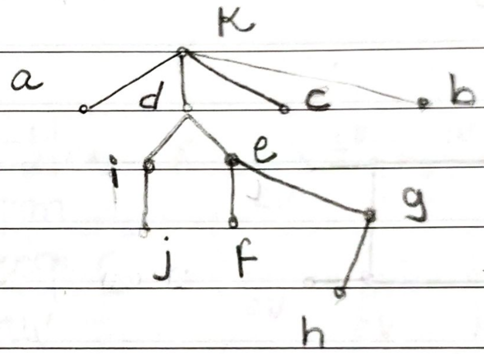
$n = 1$ $\quad$ $n = 2$ $\quad$ $n = 3$ $\quad$ $n = 4$



Para $n \geq 3$ não há árvore, pois são cíclicas.

4) Encontre o nível de cada vértice na árvore abaixo





$n = 0$

$n = 1$

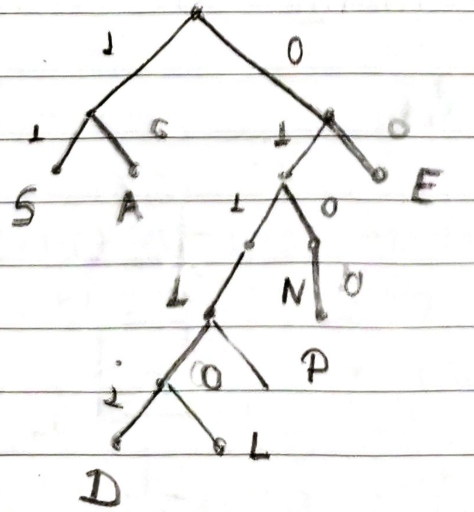
$n = 2$

$n = 3$

$n = 4$

5) Encontra a altura da árvore do exercício 4. 4 //

6) Decodifique cada sequência de bits usando o código de Huffman dado.



(a) 0110|00|010 PEN

(b) 01111|00|10|01110 DEAL

(c) 01110|10|0110 LAP

(d) 11|10|01110|10|01111 SALAD

7) Codifique cada palavra abaixo usando o mesmo código de Huffman acima.

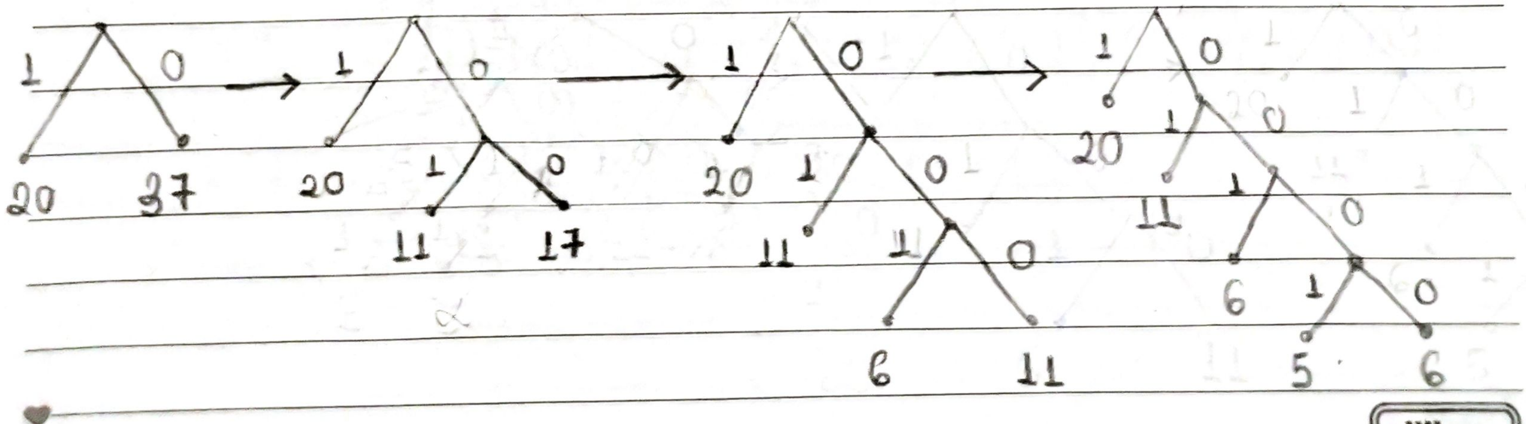
(a) DEN 01111|00|010
(b) NEED 010|00|00|01111

8) Construa um código de Huffman ótimo para o conjunto de letras da tabela.

Letter	Frequency
α	5
β	6
γ	6
δ	11
ε	20

Quanto mais comum a letra, menos bits devem ser designados à ela

5, 6, 6, 11, 20 → 5+6, 6, 11, 20
6, 11, 11, 20 → 6+11, 11, 20
11, 17, 20 → 11+17, 20
20, 37



9) Mostre que uma árvore é um conjunto bipartido

Para um grafo ser bipartido, deve haver 2 sub-conjuntos disjuntos V_1 e V_2 ($V_1 \cup V_2 = V$) onde toda aresta $e \in E$ é incidente num vértice de V_1 e num de V_2. Sabendo que todos os vértices numa árvore só formam arestas com vértices em níveis diferentes.

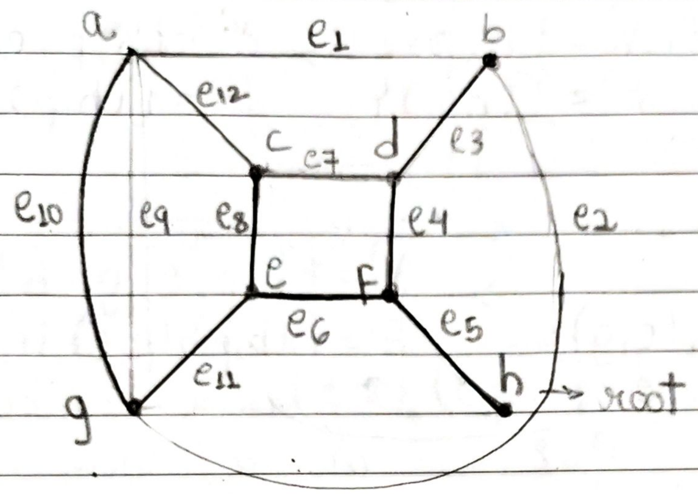
Basta que V_1 seja os vértices em níveis pares e V_2 os vértices em níveis ímpares. Dessa forma, não haverá arestas entre vértices do mesmo subconjunto.

10) Prove que T é uma árvore se, e somente se, T é conexo e quando uma aresta é adicionada, um ciclo é criado.

Por definição, uma árvore tem um único caminho de v a w para todo par de vértices v e w. Logo, para todo par de vértices existe um caminho entre eles, então T é conexo.

Se existe um caminho entre qualquer par de vértices ao adicionar uma aresta entre v e w, cria-se outro caminho de v a w. Se existem agora dois caminhos entre um par qualquer de vértices, significa que um ciclo foi criado.

11) Use o breadth-first-search com a ordem de vértices hgfedcba para encontrar uma árvore geradora para o grafo G abaixo.



f: include (d, e)

$V' = \{h\}$
$E' = \emptyset$

$V' = \{h, f\}$
$E' = \{\underbrace{(h, f)}_{e_5}\}$

$V' = \{h, f, e, d\}$
$E' = \{\underbrace{(h, f)}_{e_5}, \underbrace{(f, d)}_{e_4}, \underbrace{(f, e)}_{e_6}\}$

e: include (c, g)
d include (b)

$V' = \{h, f, e, d, b, c, g\}$
$E' = \{\underset{e_5}{(h, f)}, \underset{e_4}{(f, d)}, \underset{e_6}{(f, e)}, \underset{e_8}{(e, c)}\ \underset{e_{11}}{(e, g)}\ \underset{e_3}{(d, b)}\}$

g: include (a)
c: None
b: None

$V' = \{h, f, e, d, b, c, g, a\}$
$E' = \{(h, f), (f, d), (f, e), (e, c)\ (e, g), (d, b), (g, a)\}$

$E' = \{e_5, e_4, e_6, e_8, e_{11}, e_3, e_{10}\}$

12) Use o algoritmo Depth First Search com ordenação de vértices h g f e d c b a para o grafo em 11)

h = raiz $\quad V' = \{h\}$, $E' = \emptyset$ $\rightarrow$ $V' = \{h, f\}$, $E' = \{(h,f)\}$ $\rightarrow$ $V' = \{h, f, e\}$, $E' = \{(h,f), (f,e)\}$

$V' = \{h, f, e, g\}$
$E' = \{\underset{e_5}{(h,f)}, \underset{e_6}{(f,e)}, \underset{e_{11}}{(e,g)}\}$ $\rightarrow$ $V' = \{h, f, e, g, b\}$
$E' = \{(h,f), (f,e), (e,g), (g,b)\}$

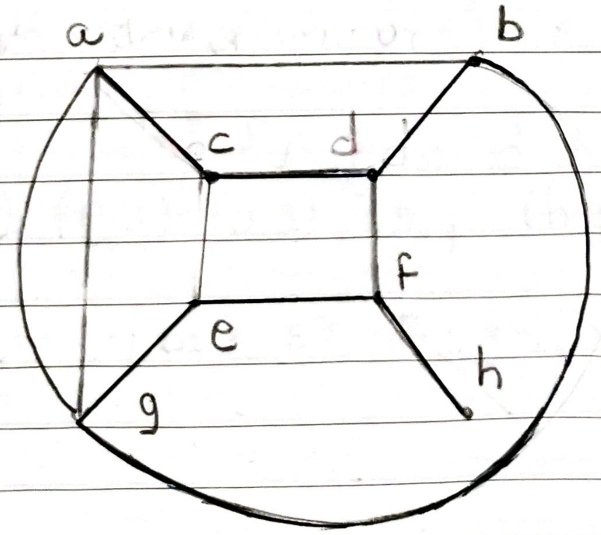
$V' = \{h, f, e, g, b, d\}$
$E' = \{(h,f), (f,e), (e,g), (g,b), (b,d)\}$

$\downarrow$

$V' = \{h, f, e, g, b, d, c\}$
$E' = \{(h,f), (f,e), (e,g), (g,b), (b,d), (d,c)\}$

$\downarrow$

$V' = \{h, f, e, g, b, d, c, a\}$
$E' = \{(h,f), (f,e), (e,g), (g,b), (b,d), (d,c), (c,a)\}$



16) Sob quais condições uma aresta em um grafo G estará em qualquer árvore geradora de G.

Ela está em qualquer árvore geradora se sua remoção torna G desconexo.

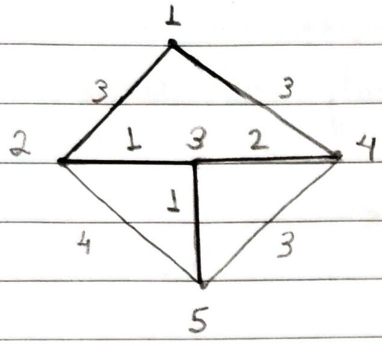
Seja e uma aresta em G e G' é o grafo G sem a aresta e (desconexo)
Seja T uma árvore geradora de G. Vamos assumir que T não contém e. Visto que T não contém e, T também é uma árvore geradora para G'. Contudo, se T é conexo, G' deveria ser conexo também, o que é uma contradição. Logo, T deve conter e.

17) Sejam T e T' duas árvores geradoras de um grafo conexo G. Suponha que uma aresta x está em T mas não em T'. Mostre que existe uma aresta y em T' mas não em T, t.q $(T - \{x\}) \cup \{y\}$ e $(T' - \{y\}) \cup \{x\}$ são árvores geradoras de T.

Suponha que x incide nos vértices a e b. Removendo x de T produz um grafo desconexo de dois componentes U e V. Os vértices a e b pertencem a componentes diferentes, digamos $a \in U$, $b \in V$. Existe um caminho P de a para b em T. À medida que nos movemos ao longo de P, em algum ponto encontramos uma aresta $y = (v, w)$ com $v \in U$, $w \in V$. Visto que somar y a $T - \{x\}$ produz um grafo conexo, então, $(T - \{x\}) \cup \{y\}$ é uma árvore geradora. Claramente, $(T' - \{y\}) \cup \{x\}$ é uma árvore geradora.

19) Nos itens abaixo, encontre a árvore geradora minimal (MST) dada pelo algoritmo Prim para cada grafo.

(a)



· Se arestas tem pesos iguais, é possível que a MST não seja única

$v(s) = 1$
$v(i) = 0$
$E = \emptyset$

aresta	peso
(1,2)	3 ←
(1,4)	3

⇓

aresta	peso
(1,4)	3
(2,3)	1 ←
(2,5)	4

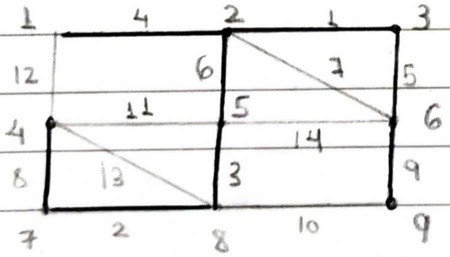
⇐

aresta	peso
(1,4)	3
(2,5)	4
(3,4)	2
(3,5)	1 ←

aresta	peso
(1,4)	3
(2,5)	4
(3,4)	2 ←
(5,4)	3

$E' = \{(1,2), (2,3), (3,5), (3,4)\}$

(b)



aresta	peso
(1,2)	4 ←
(1,4)	12

⇓

aresta	peso
(1,4)	12
(2,3)	1 ←
(2,6)	7
(2,5)	6

⇐

aresta	peso
(1,4)	12
(2,6)	7
(2,5)	6
(3,6)	5 ←

aresta	peso
(1,4)	12
(2,6)	7
(2,5)	6 ←
(6,9)	9

⇓

aresta	peso
(1,4)	12
(6,9)	9
(5,4)	11
(5,8)	3 ←

⇒

aresta	peso
(1,4)	12
(6,9)	9
(8,7)	2 ←
(8,9)	10

aresta	peso
(1,4)	12
(6,9)	9
(8,9)	10
(7,4)	8 ←

aresta	
(6,9)	9 ←
(8,9)	10

E'
(1,2)
(2,3)
(3,6)
(6,9)
(2,5)
(5,8)
(8,7)
(7,4)