

UENF

Universidade Estadual do Norte Fluminense Darcy Ribeiro

Curso: Ciência de Computação

Data: 13./11./2023

Prova: Questões P2

Período: 4º

Disciplina: Estrutura de dados II

Professor: Fermín Alfredo Tang

Turno: Diurno

Nome do Aluno:**Matrícula:**

1. [2,0 Pontos] Crie uma árvore AVL usando os seguintes dados ingressados em sequência:

80	70	66	56	33	23	14	10	7
----	----	----	----	----	----	----	----	---

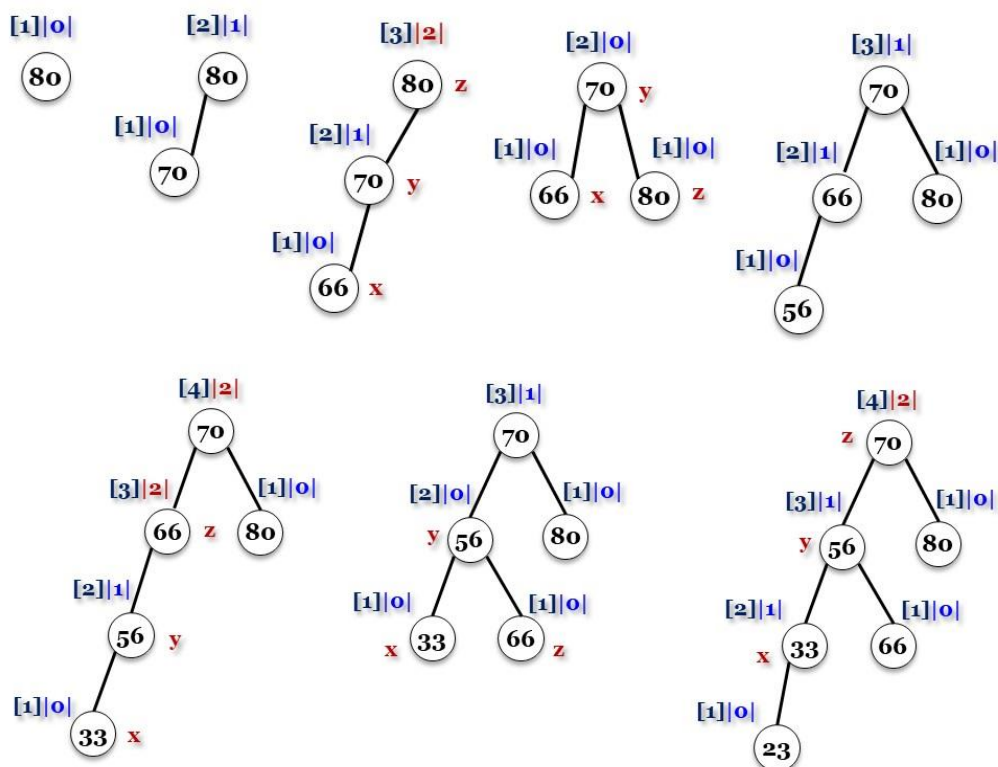
- i) Desenhe a árvore após cada inserção. Indicando a altura de cada nó e o fator de balanceamento. Caso seja necessário realize o balanceamento da árvore, indicando o tipo de rotação utilizada.

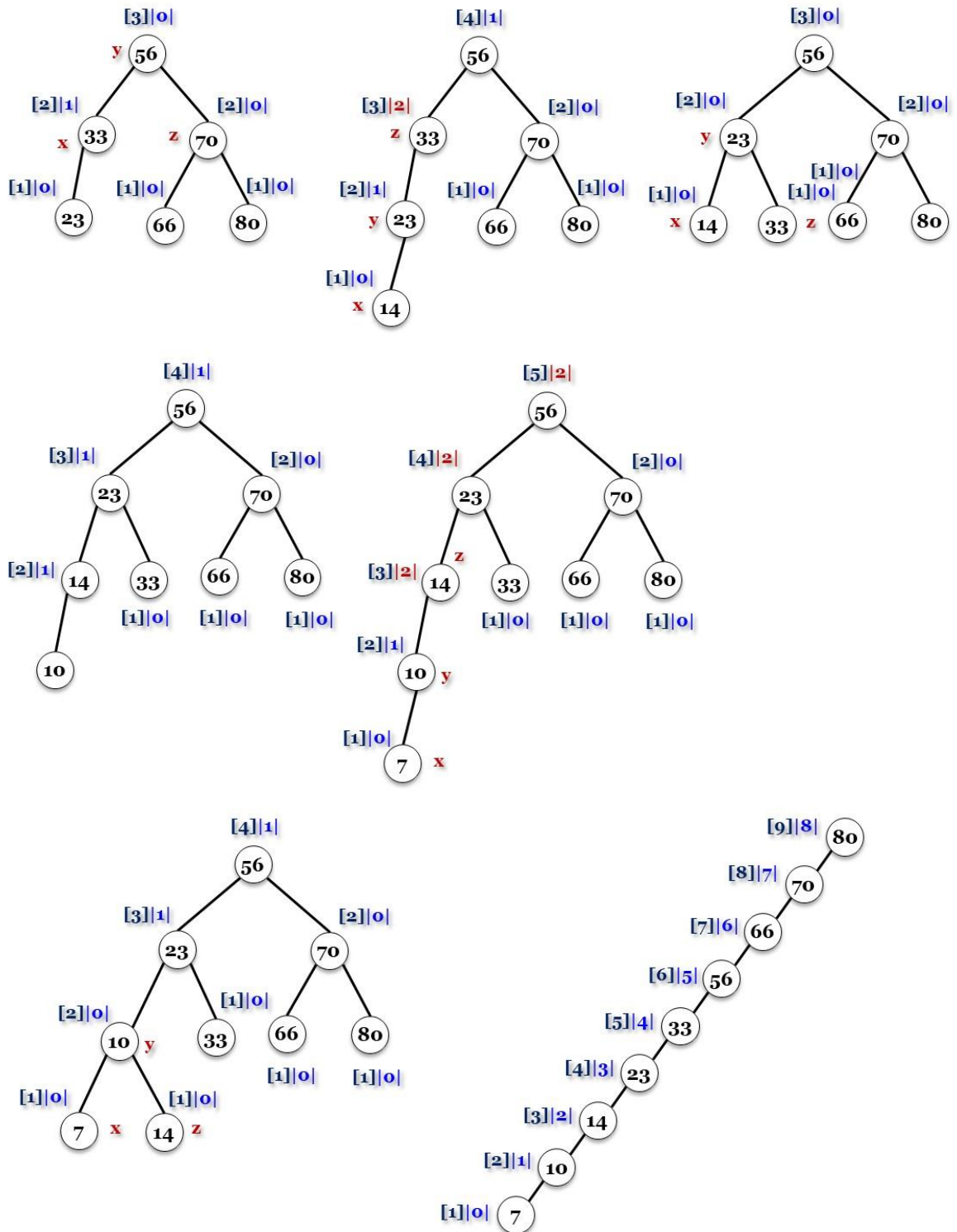
(0,2 ponto

c/u).

- ii) Compare a altura da árvore AVL final com a árvore de pesquisa binária não balanceada. (0,2 ponto).

Resposta 1.-





2. [2,2 Pontos] Crie uma árvore AVL usando os seguintes dados ingressados em sequência:

14 23 7 10 33 56 80 66 70

- iii) Desenhe a árvore após cada inserção. Caso necessário realize o balanceamento indicando o tipo de operação realizada. (0,2 ponto cada inserção).
- iv) Mostre os fatores de balanceamento na árvore final (0,4 ponto).

3. [2,0 Pontos] Responda as seguintes questões sobre árvores, justificando a sua resposta:

(0,5 ponto c/u).

- i) Usando as definições de *Heap* e árvore AVL, no pior, caso qual dessas árvores teria maior altura para a mesma quantidade de nós n . Justifique a sua resposta.
- ii) Qual é o número máximo de rotações (sejam simples ou duplas) necessárias em uma árvore AVL após uma inserção?.
- iii) Embora um *Heap* seja uma árvore ele é costuma ser implementado usando um vetor. Considerando as operações de inserção e remoção em um Heap, explique porque isso é possível?.
- iv) O objetivo do balanceamento em uma árvore AVL é ganhar eficiência no acesso aos dados armazenados. Explique qual seria o impacto de não realizar o balanceamento.

Resposta 3.-

- i) No pior caso a árvore AVL teria maior altura, uma vez que esse tipo de árvore permite uma diferença de altura entre as subárvores esquerda e direita de até 1. Sendo que essa diferença é válida entre as subárvores de qualquer nível. Já *Heap* é uma árvore compacta na qual somente se permite tal diferença entre as subárvores da raiz.
- ii) Somente será necessária uma rotação (Seja simples ou dupla).
- iii) O *Heap* é uma estrutura que representa uma árvore quase completa, na qual apenas o último nível pode está incompleto. Sendo que o último nível é sempre preenchido de esquerda a direita. Ele pode ser representado por um vetor, que é preenchido de acordo com o percurso em largura da árvore, de maneira que os elementos do *Heap* possuem posições fixas no vetor. No caso da inserção, o novo elemento é sempre inserido no último nível ou na última posição do vetor. No caso da remoção, o elemento raiz é removido por substituição, copia-se o último nó do último nível na raiz, e elimina-se o último nó no último nível ou na última posição do vetor.
- iv) Caso o balanceamento de uma árvore binária não fosse realizado, no pior caso a árvore se tornaria uma lista linear, fazendo com que o custo de acesso, inserção e remoção fosse de ordem $O(n)$, enquanto que quando balanceada seria aproximadamente $\log(n)$.

4. [2,0 Pontos] Considere a árvore vermelho-preta mostrada na Figura 1:

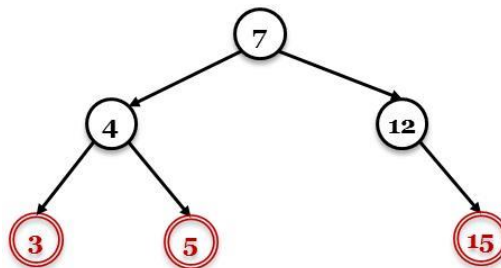
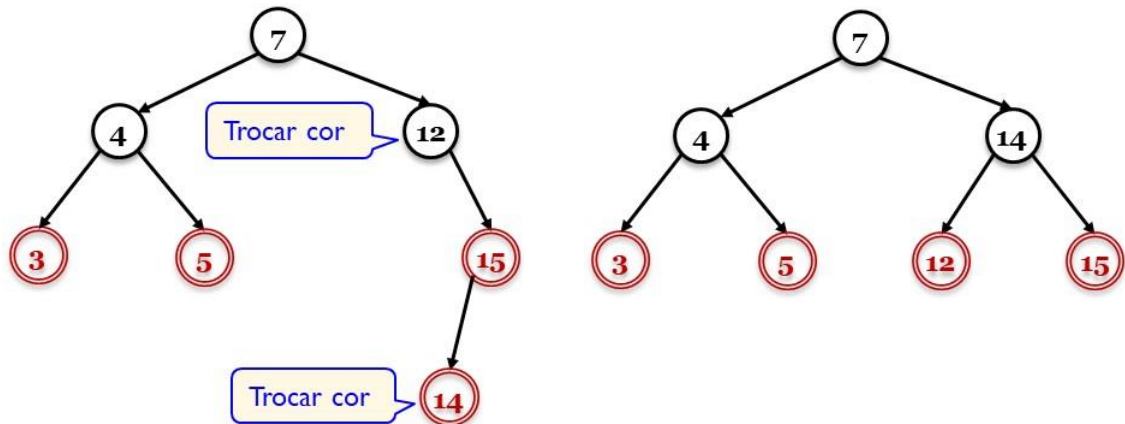


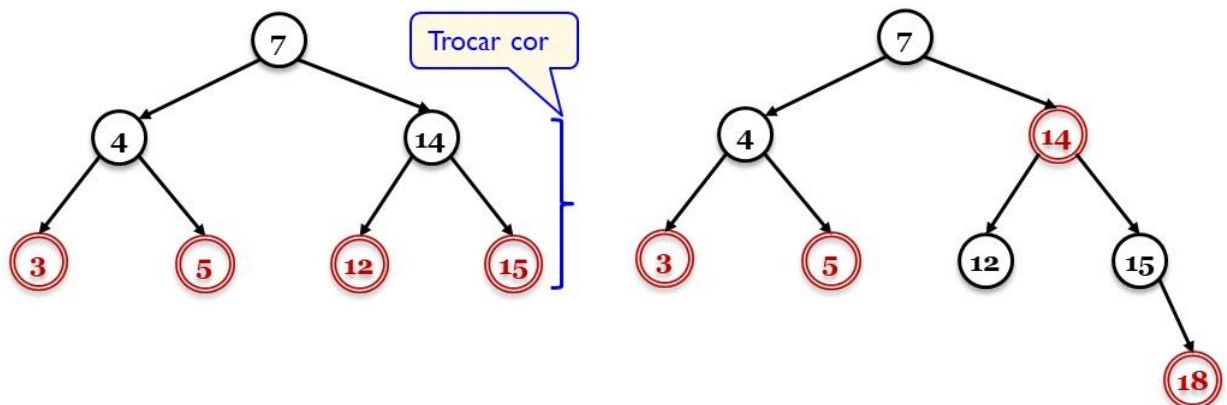
Figura 1. Árvore vermelho-preta

Realize a inserção da seguinte sequência de nós: 14, 18, 16, 17. Desenhe a árvore resultante após cada inserção. Indicando os procedimentos de balanceamento realizados. [0,5 ponto cada inserção].

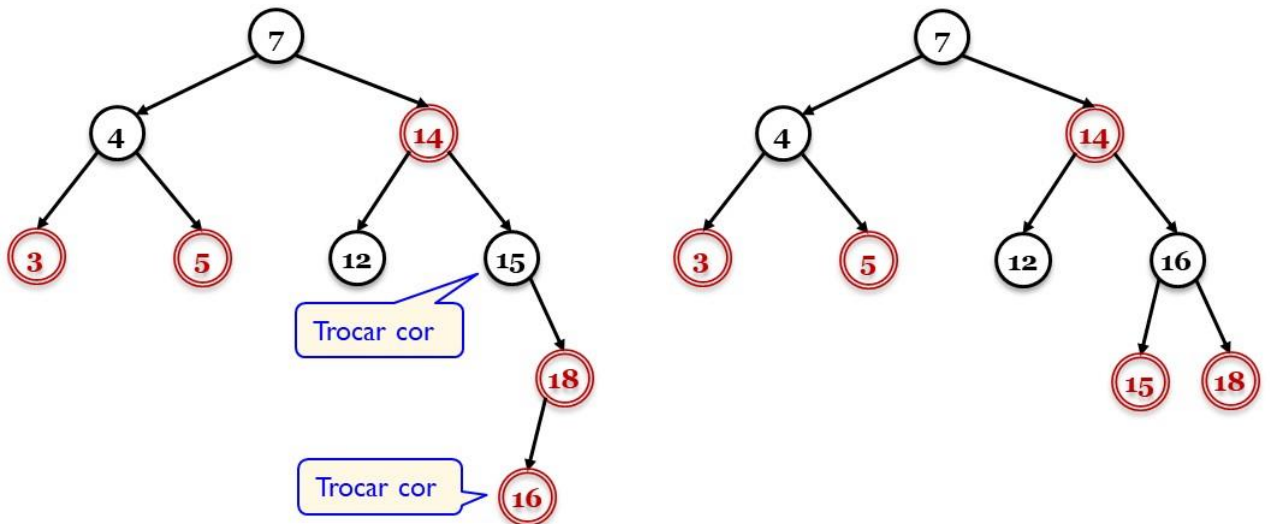
Resposta4.- A inserção da chave 14 viola a restrição de não poder ter dois nós vermelhos consecutivos. Para restaurar a condição vermelho-preta, identifica-se o caso, e realiza-se uma troca de cores e uma rotação dupla.



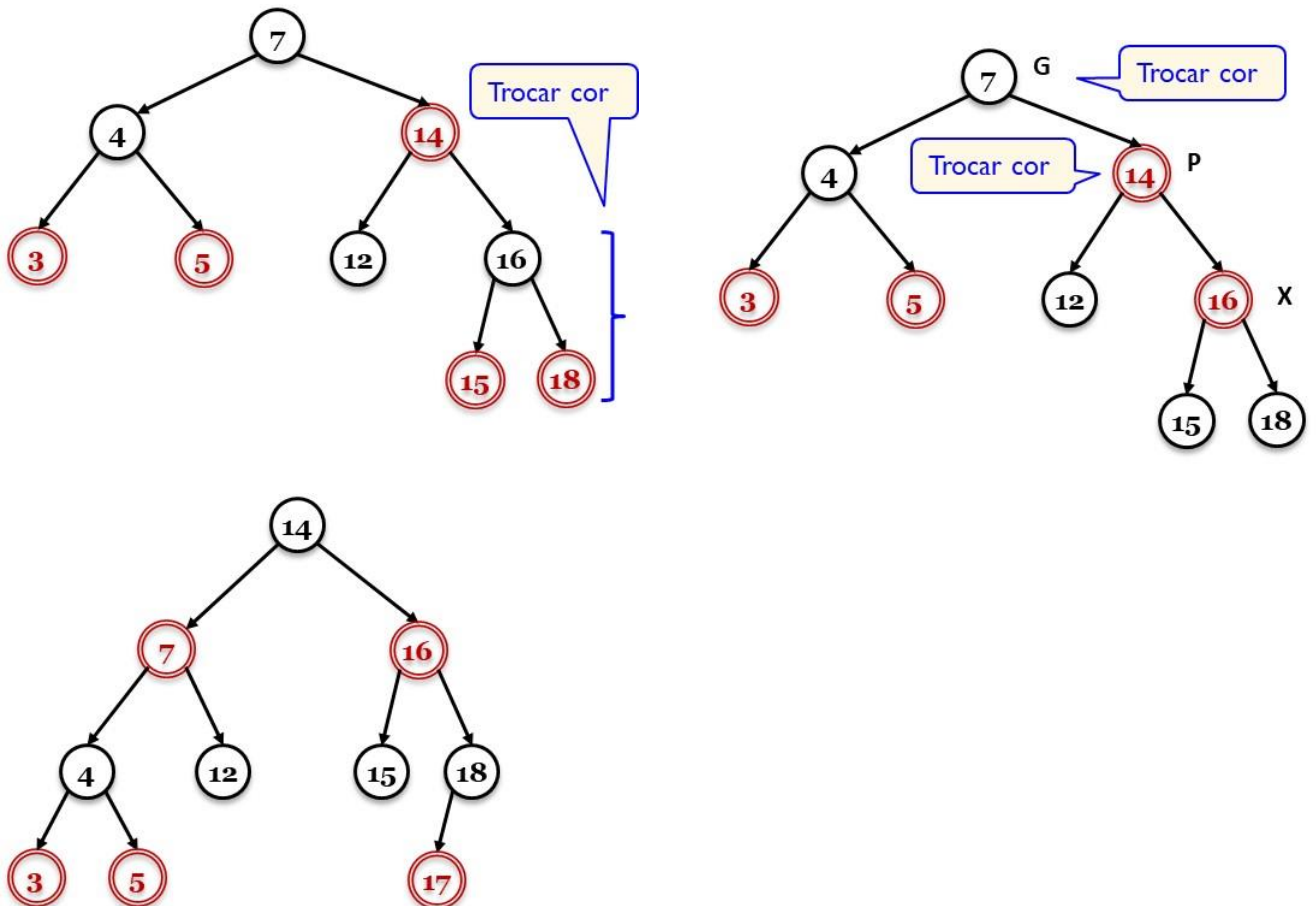
Na inserção da chave 18, durante a descida identifica-se o padrão, pai preto, dois filhos vermelhos. Neste caso, realiza-se uma alternância de cores, o que viabiliza a inserção.



A inserção da chave 16 viola a restrição de não poder ter dois nós vermelhos consecutivos. Para restaurar a condição vermelho-preta, identifica-se o caso, e realiza-se uma troca de cores e uma rotação dupla direita-esquerda.



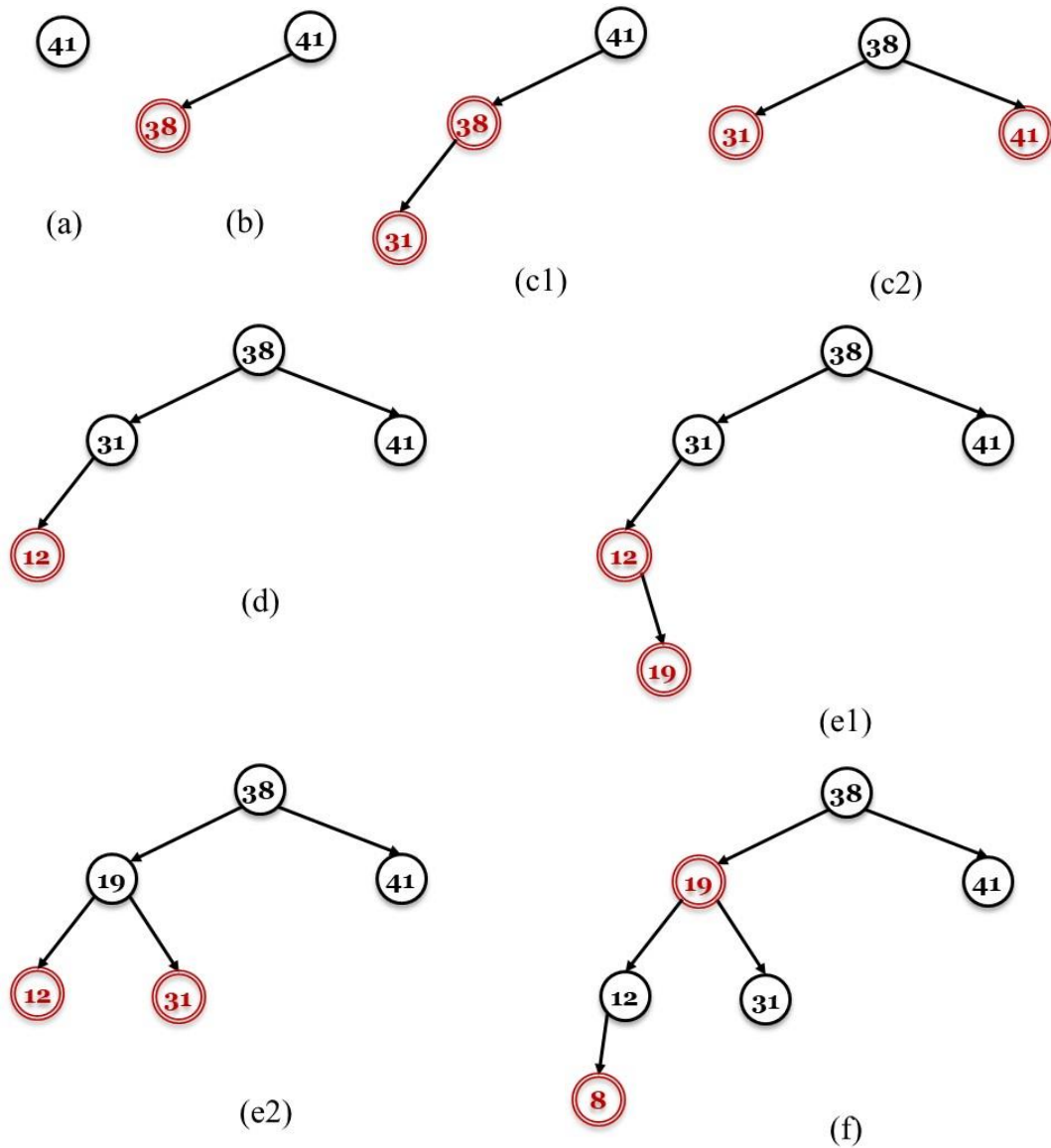
Na inserção da chave 17, durante a descida identifica-se o padrão, pai preto, dois filhos vermelhos. Neste caso, realiza-se uma alternância de cores, no entanto esta alternância produz uma violação do tipo pai vermelho, filho vermelho. **Realiza-se uma rotação durante a descida.** Neste caso uma rotação esquerda simples com as respectivas trocas de cor. Finalmente insere-se a chave 17.



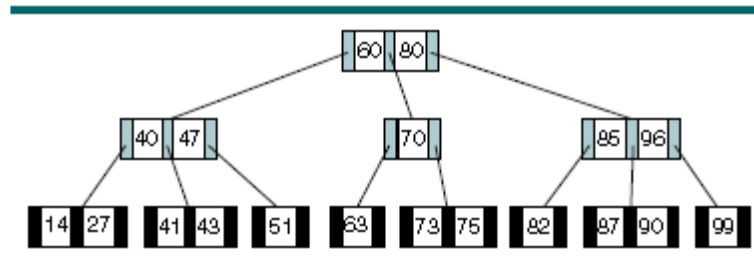
5. [1,8 Pontos] Mostre a árvore vermelha-preta que resulta após inserir os nós com chaves 41, 38, 31, 12, 19, 8 em uma árvore vermelha-preta inicialmente vazia.

Desenhe a árvore resultante após cada inserção. Indicando os procedimentos de balanceamento realizados. [0,3 ponto cada inserção].

Resposta 5.-



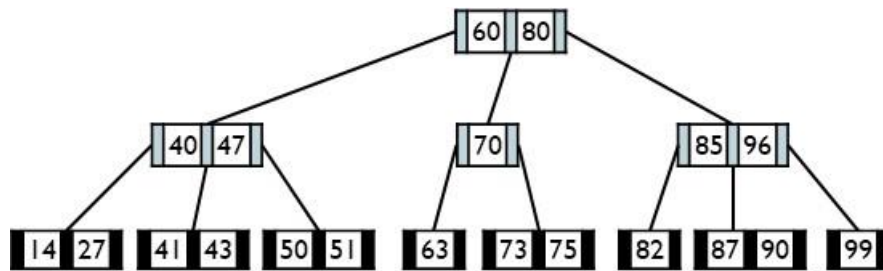
6. [2,0 Pontos] Usando a árvore B de ordem 3, mostrada na figura embaixo:



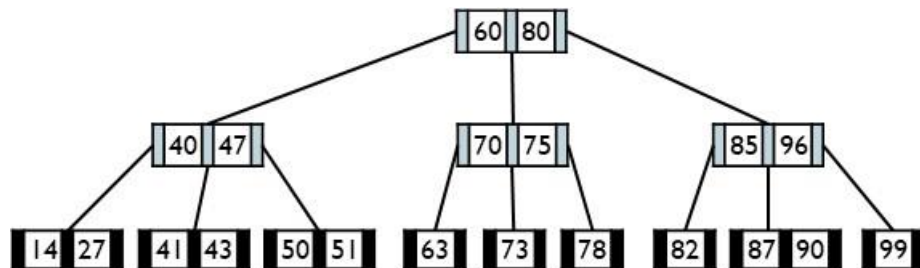
Adicione 50, 78, 101 e 232. Desenhe a árvore após cada inserção e explique o que acontece em cada caso. [0,5 ponto cada inserção].

Resposta 6.-

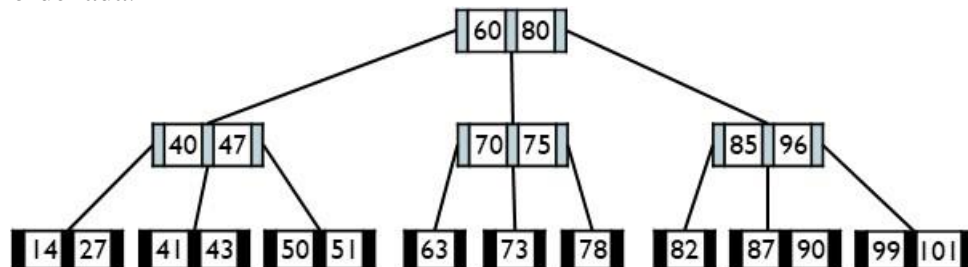
A chave 50 encontra espaço disponível no nó destino e é inserida na posição ordenada.



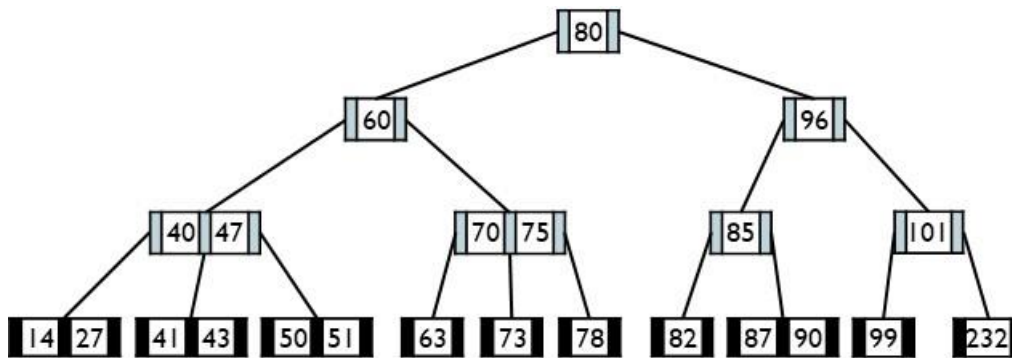
A chave 78 encontra o seu nó destino cheio, com isso se procede a uma divisão desse nó, reconduzindo a chave intermediária, chave 75 a ser realocado no nó pai, que possui espaço disponível.



A chave 101 encontra o seu nó destino com espaço disponível e é inserida na posição ordenada.



A chave 232 encontra o seu nó destino cheio, com isso se procede a uma divisão desse nó, reconduzindo a chave intermediária, chave 101 a ser realocado no nó pai. Como o nó pai não possui espaço disponível, acontece nova divisão, agora do nó pai. A chave intermediária 96 é realocada para no nó pai acima. Uma terceira divisão é requerida, nesta vez no nó raiz. Cria-se uma nova raiz que armazena a chave intermediária 80. A cada divisão os ponteiros são redefinidos de maneira adequada.



7. [2,2 Pontos] Descreva as características de uma árvore B de ordem m , considerando a organização dos nós, número de entradas em cada nó, número de filhos, balanceamento, taxa de ocupação da árvore [0,3 cada tópico].

Elabora um exemplo de árvore B de ordem $m = 3$, com 3 níveis. [0,7 pontos].

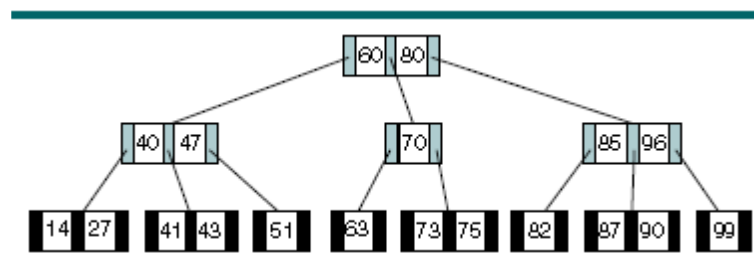
Resposta 7.- Uma árvore B de ordem m é uma **árvore balanceada** que admite que seus nós possuam várias entradas ou chaves. As chaves contidas em cada nó estão organizadas em ordem crescente. Além de possuir várias entradas ou chaves, cada nó pode possuir vários nós filhos, dispostos de esquerda a direita e contendo chaves de menor valor à esquerda e maior valor à medida que avança para direita. A organização dos nós pode ser considerada uma generalização das árvores binárias para vários filhos.

Em geral, um nó mais à esquerda contém chaves menores que a menor chave do seu pai imediato. Um nó mais à direita contém chaves maiores que a maior chave do seu pai imediato. Já um nó intermediário contém chaves cujos valores se encontram entre os valores de duas chaves armazenadas consecutivamente no nó pai.

A ordem m refere-se ao **número máximo de filhos** que um nó pode ter. Sendo que o **número máximo de entradas** em cada nó corresponde a $e = m - 1$.

Na árvore B, espera-se que cada nó possua **aproximadamente metade das entradas preenchidas**. A única exceção é o nó raiz, que pode possuir um mínimo de 1 entrada e dois filhos. Nesta árvore, todos os níveis estão necessariamente completos.

Um exemplo deste tipo de árvore é o seguinte:



8. [1,55 Pontos] Percorra o grafo da Figura 2 começando pelo nó 1 e usando **busca em profundidade**. Mostre a árvore de busca e a estrutura de dados utilizada nas decisões.

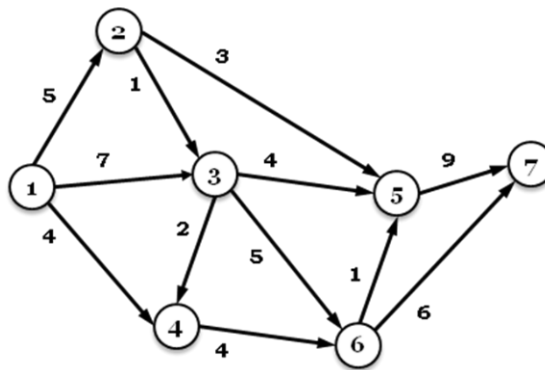
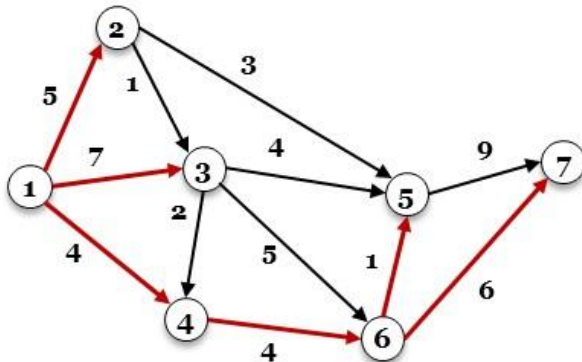


Figura 1. Percurso em profundidade

Resposta 8.- A árvore de busca em profundidade é a seguinte:



Pilha	Visitados	Pilha	Visitados	Pilha	Visitados	Pilha	Visitados
4	1	4	1	4	1	4	1
3	0	3	1	3	1	3	1
2	0	2	1	2	1	2	1
1	0	1	1	1	1	1	1
	0		0		0		1
	0		0		1		1
	0		0		0		1

Pilha	Visitados	Pilha	Visitados	Pilha	Visitados	Pilha	Visitados
3	1	3	1	3	1	3	1
2	1	2	1	2	1	2	1
1	1	1	1	1	1	1	1
	1		1		1		1
	1		1		1		1
	1		1		1		1
	1		1		1		1

9. [1,55 Pontos] Percorra o grafo da Figura 2 começando pelo nó 1 e usando **busca em largura**. Mostre a árvore de busca e a estrutura de dados utilizada nas decisões.

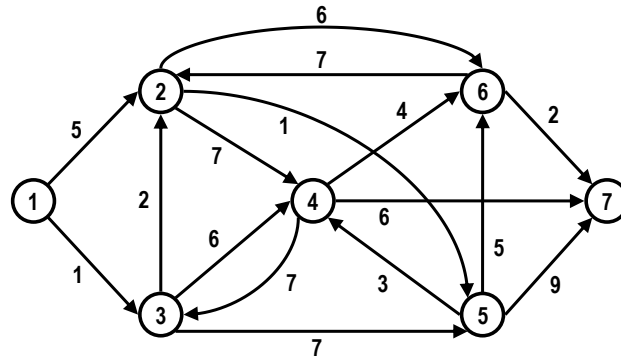
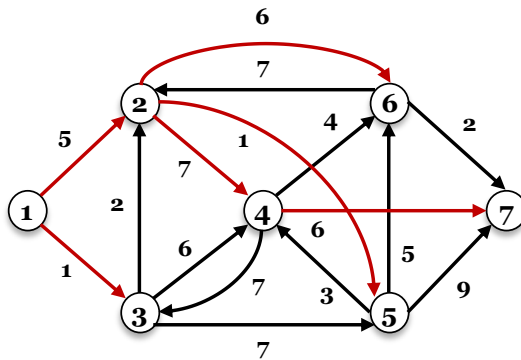


Figura 2. Percurso em Largura

Resposta 9.- A busca em largura utiliza uma estrutura de fila para armazenar os nós descobertos e processá-los. Começando a busca no nó 1, temos a seguinte árvore de busca, com as descobertas e processamento de nós na fila, e os nós visitados:



	1	2	3	4
Fila	1			
Fila	2	3		
Fila	3	4	5	6
Fila	4	5	6	
Fila	5	6	7	
Fila	6	7		
Fila	7			

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