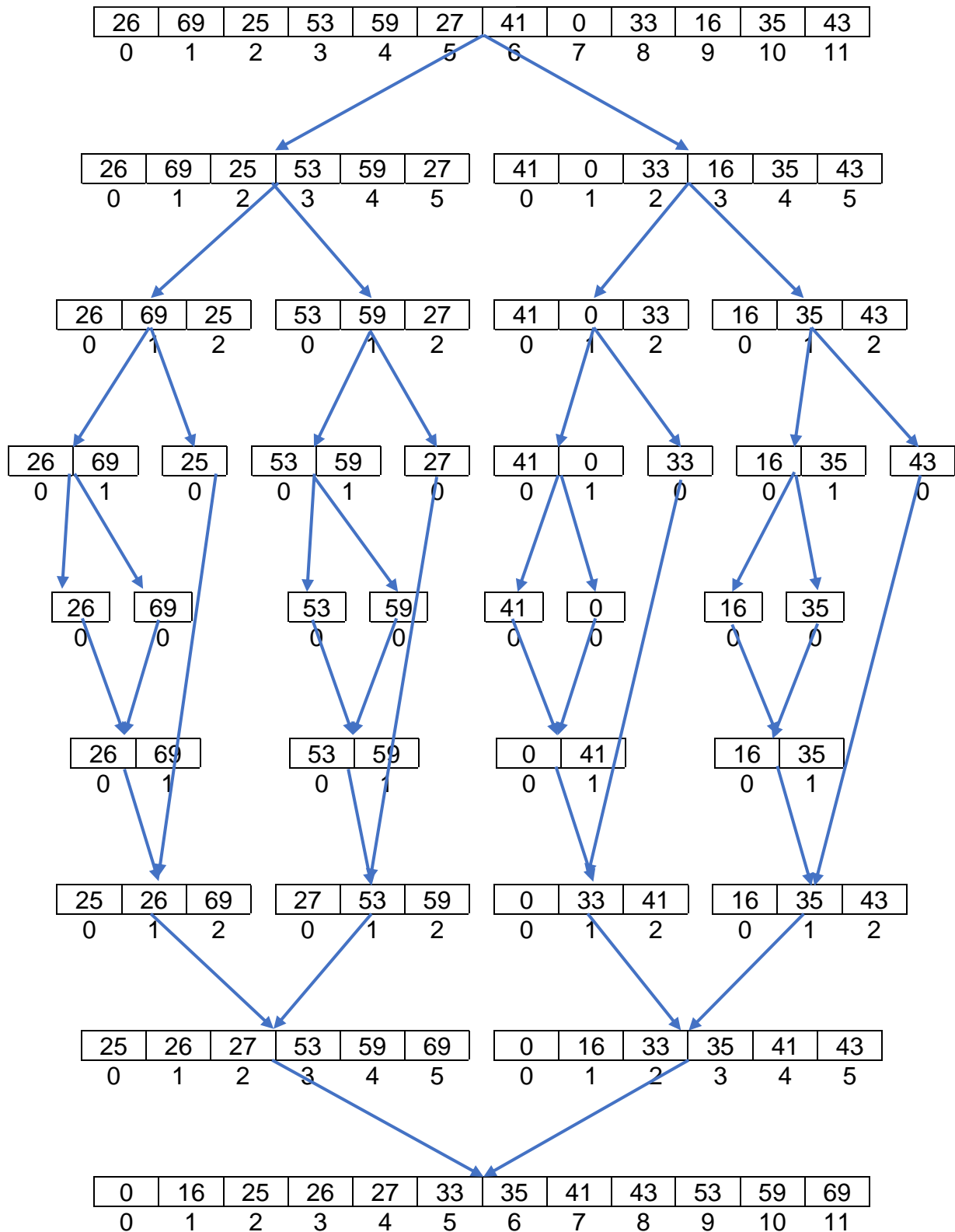


**Lista 8 – Revisão Lista Duplamente Encadeada, Quick Sort, Merge Sort, Heap Sort, Árvores e Grafos**

1) Realize a simulação com o Algoritmo Merge Sort até que ocorra a ordenação completa do Vetor Abaixo:  
Considere o vetor abaixo:

26	69	25	53	59	27	41	0	33	16	35	43
----	----	----	----	----	----	----	---	----	----	----	----



4) Realize a ordenação do algoritmo Quick Sort até que ocorra a ordenação completa do vetor abaixo:

PIVÔ=99:

149	192	47	152	159	195	61	66	17	167	118	64	27	80	30	105
-----	-----	----	-----	-----	-----	----	----	----	-----	-----	----	----	----	----	-----

99	142	192	47	152	159	195	61	66	17	167	118	64	27	80	30	105
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

47	61	66	17	64	27	80	30	99	142	192	152	169	195	167	118	105
0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7

17	27	30	47	61	66	64	80	99	118	105	142	192	152	169	195	167
0	1	2		0	1	2	3		1	2		0	1	2	3	4

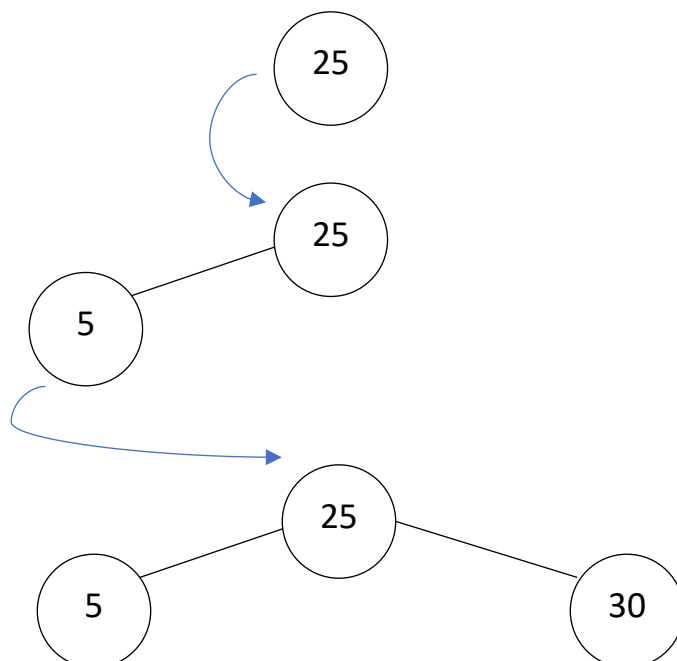
17	27	30	47	61	66	64	80	99	105	118	142	152	169	167	192	195
	1	2			0	1	2					0	1	2		

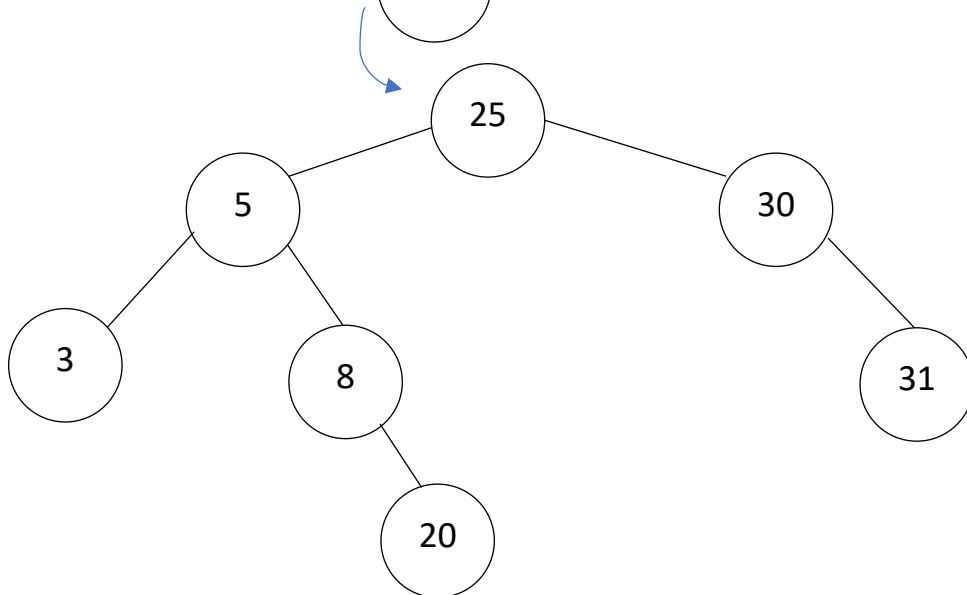
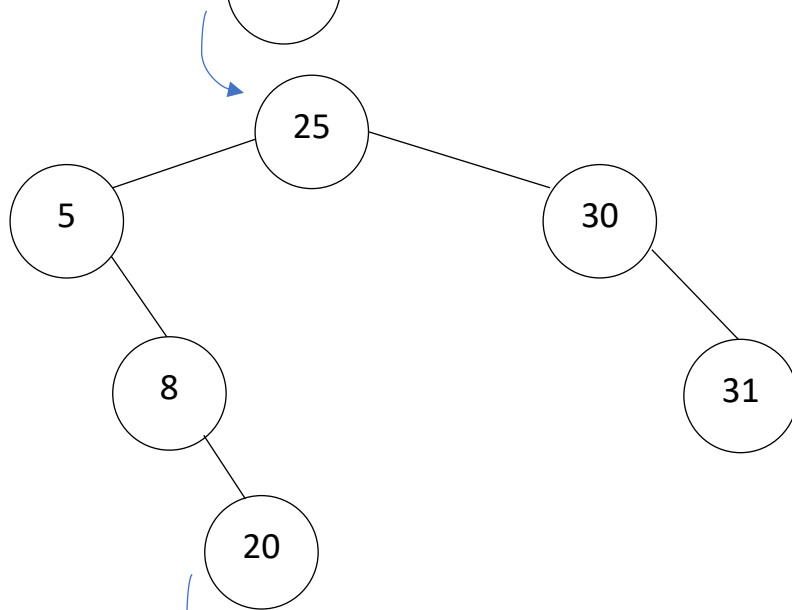
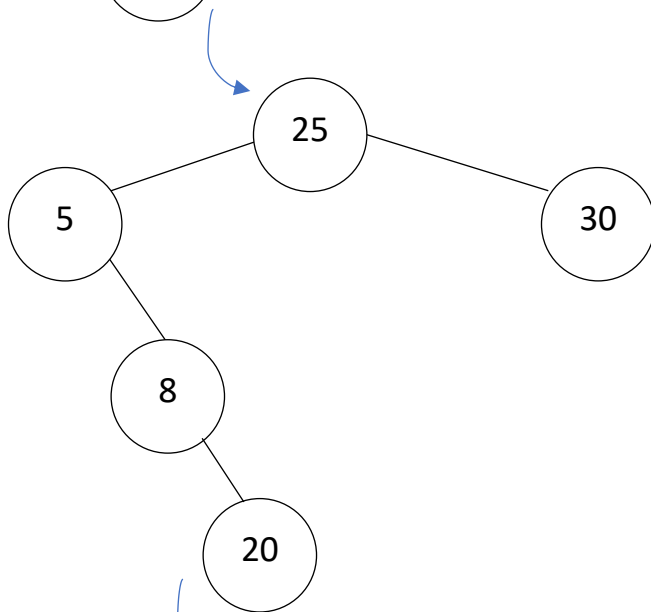
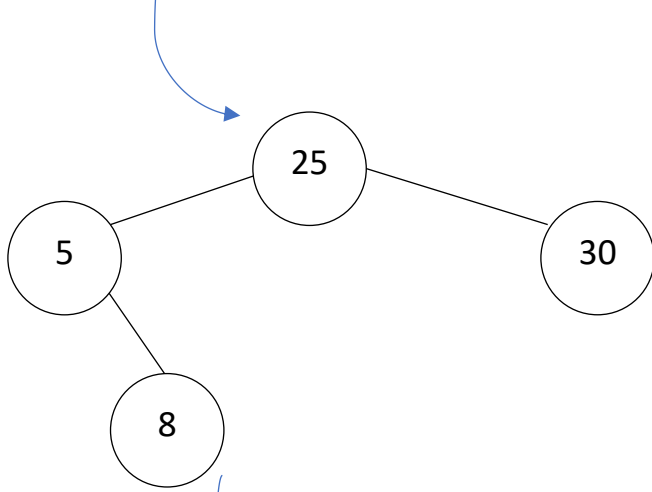
17	27	30	47	61	64	66	80	99	105	118	142	152	169	167	192	195
													0	1		

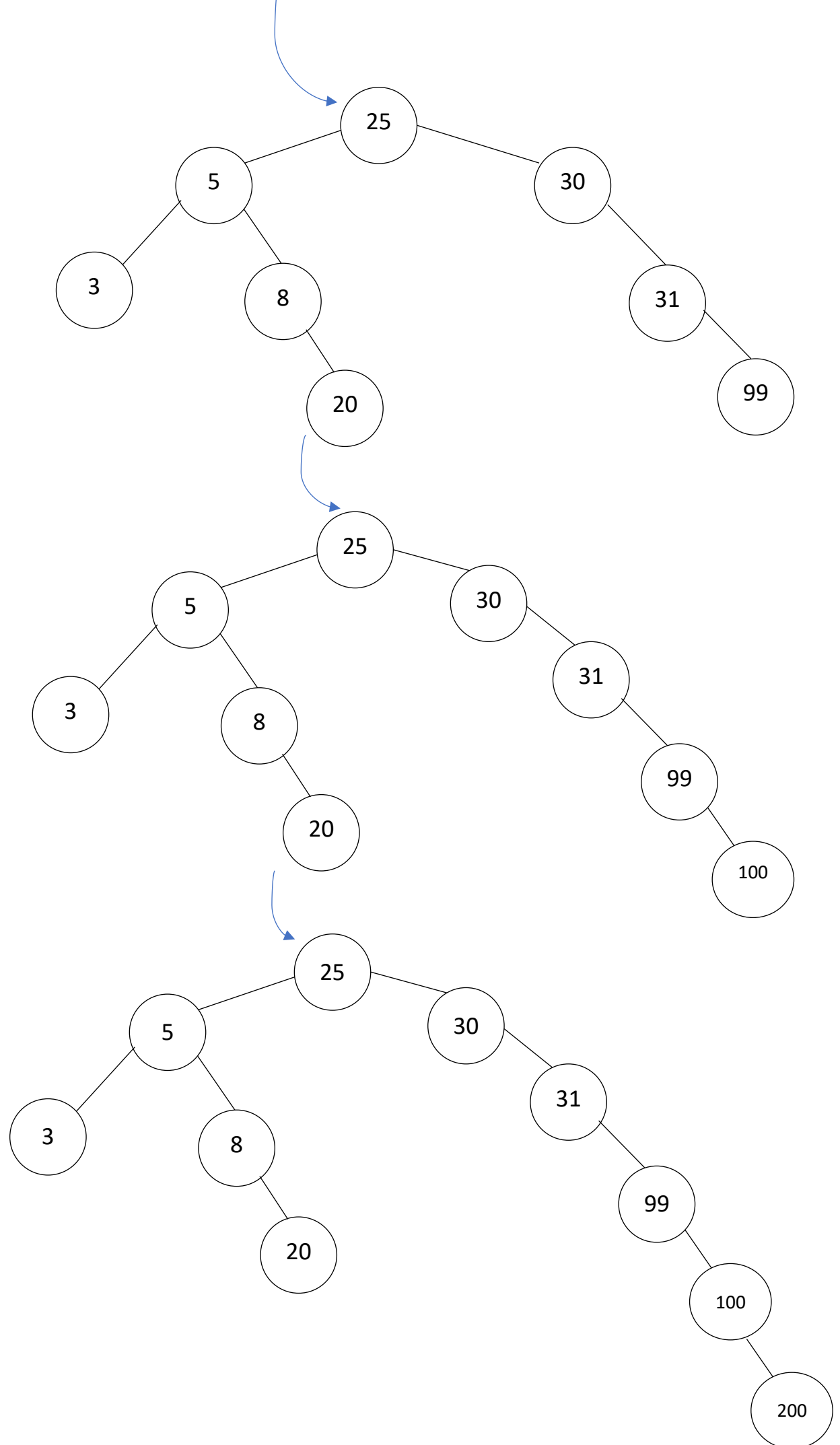
17	27	30	47	61	64	66	80	99	105	118	142	152	167	169	192	195
----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----

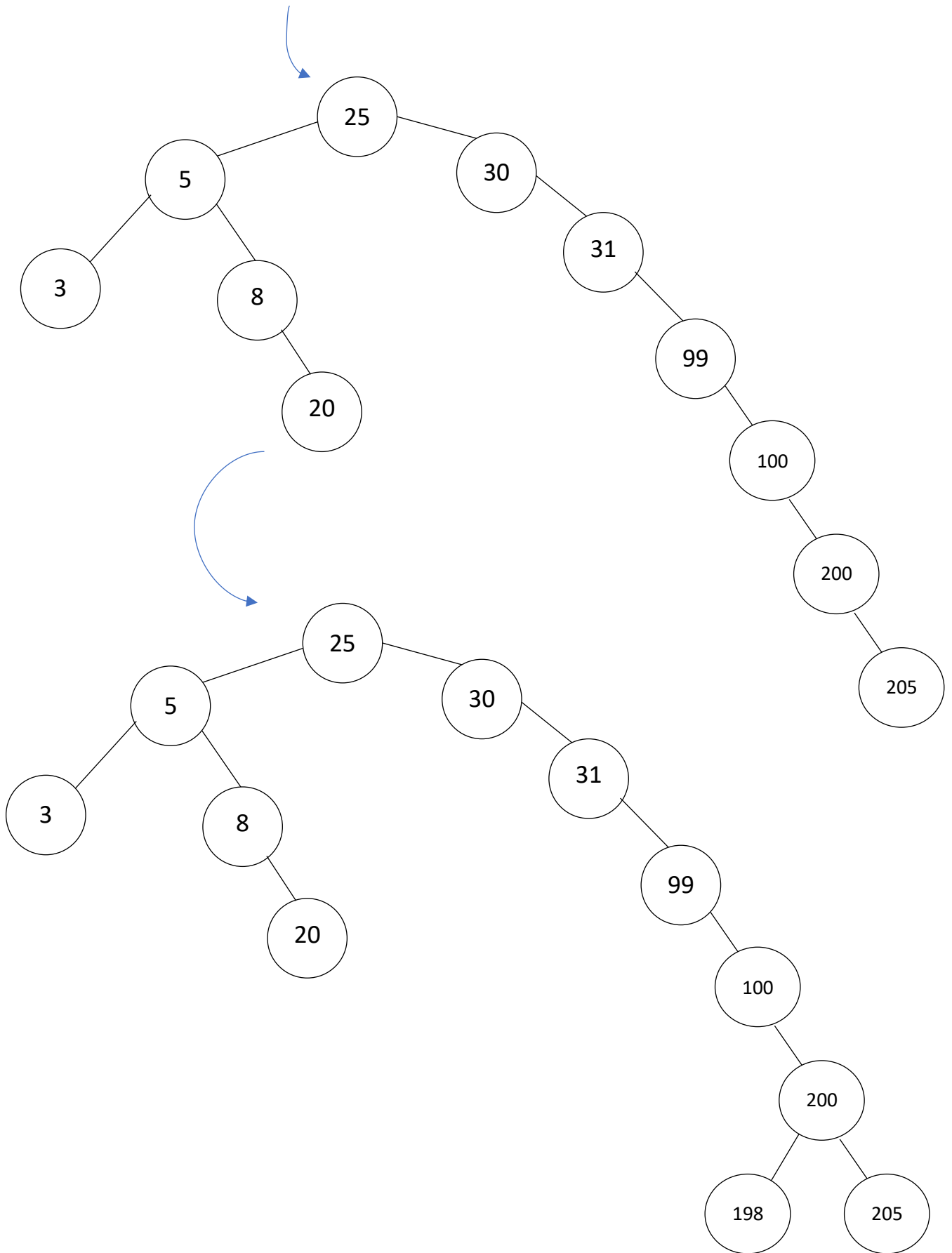
17	27	30	47	61	64	66	80	99	105	118	142	152	167	169	192	195
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

7) Simule todos os passos até o estado final da árvore binária para os seguintes elementos: 25, 5, 30, 8, 20, 31, 3, 99, 100, 200, 205, 198. Explique quais tipos de percursos poderiam ser usados para a árvore.



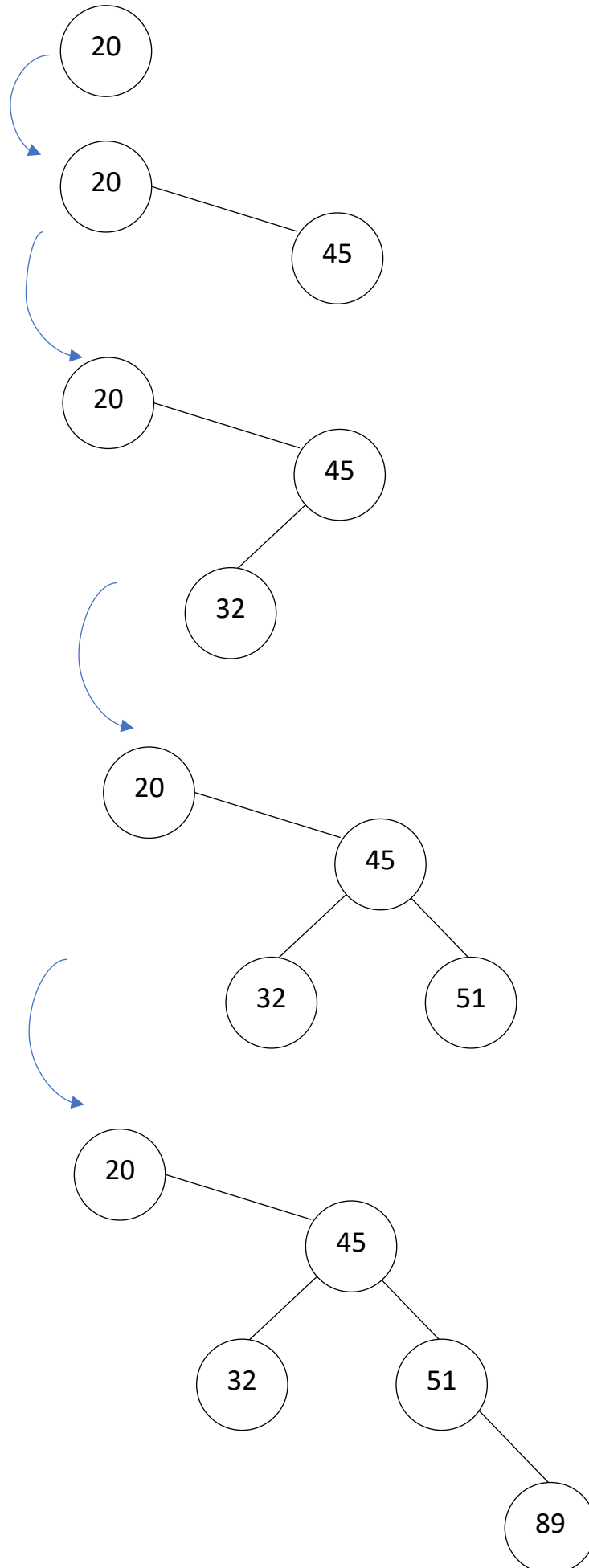


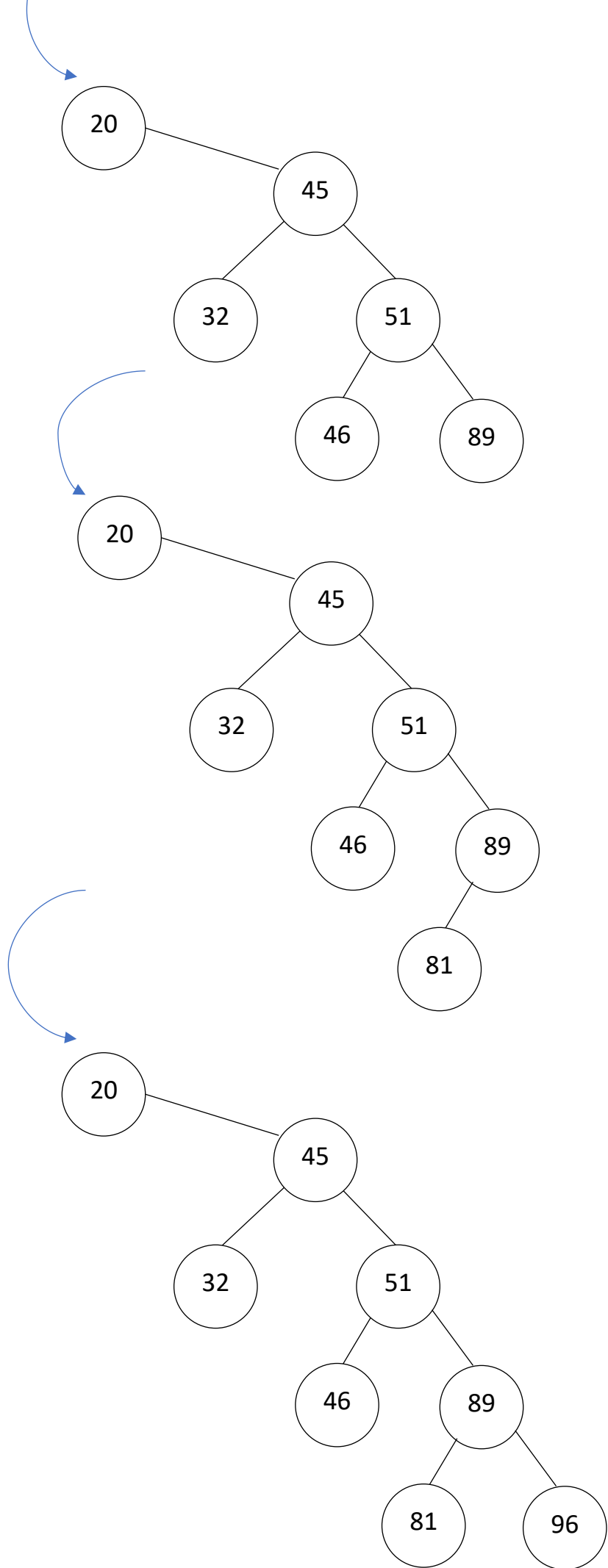


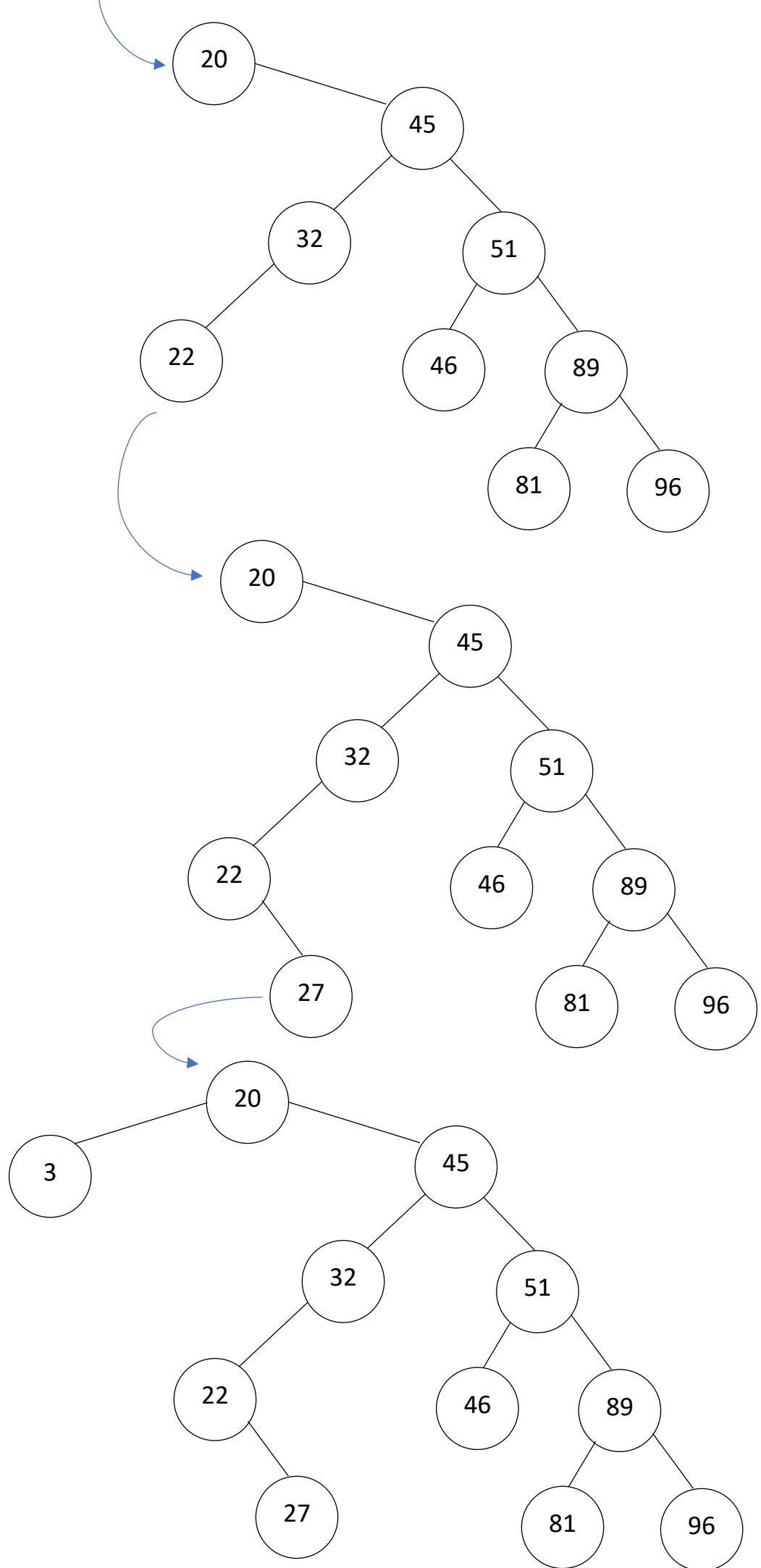


Pré-ordem: [25, 5, 3, 8, 20, 30, 31, 99, 100, 200, 198, 205]  
Em ordem: [3, 5, 8, 20, 25, 30, 31, 99, 100, 198, 200, 205]  
Pos ordem: [3, 20, 8, 5, 198, 205, 200, 100, 99, 31, 30, 25]

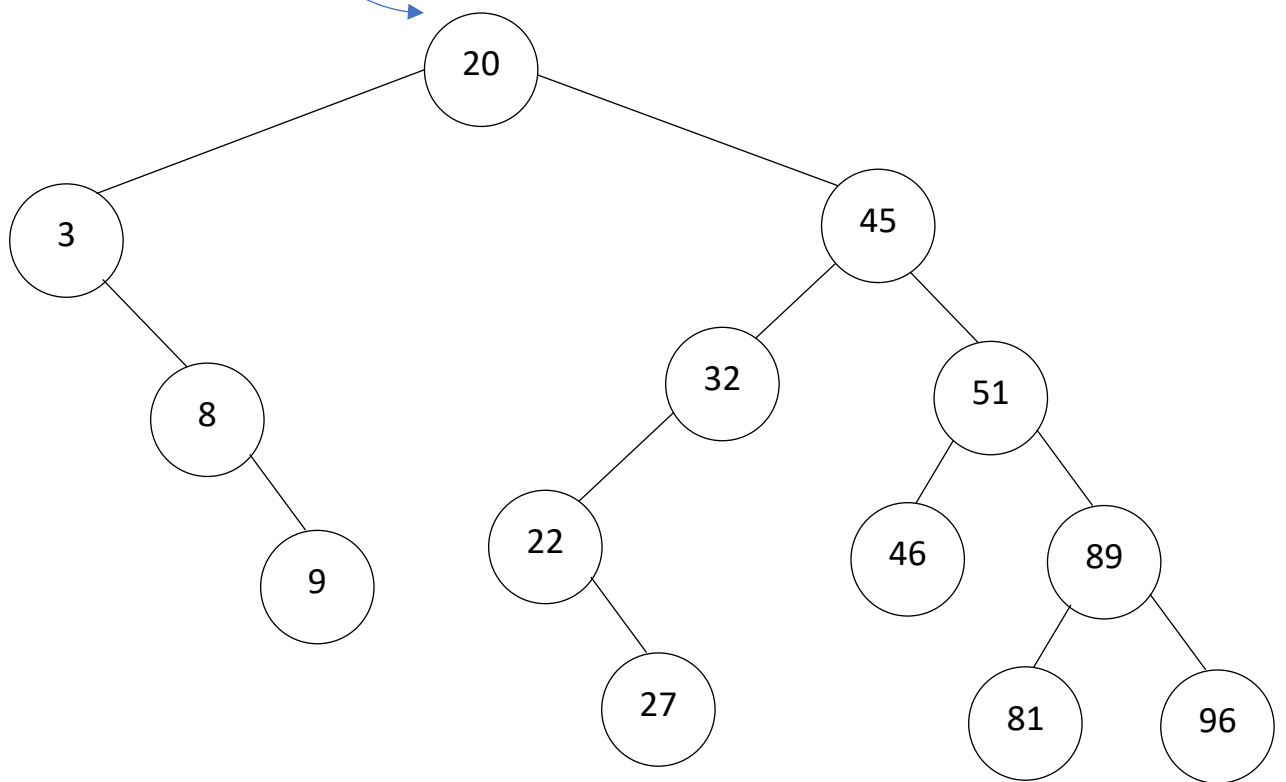
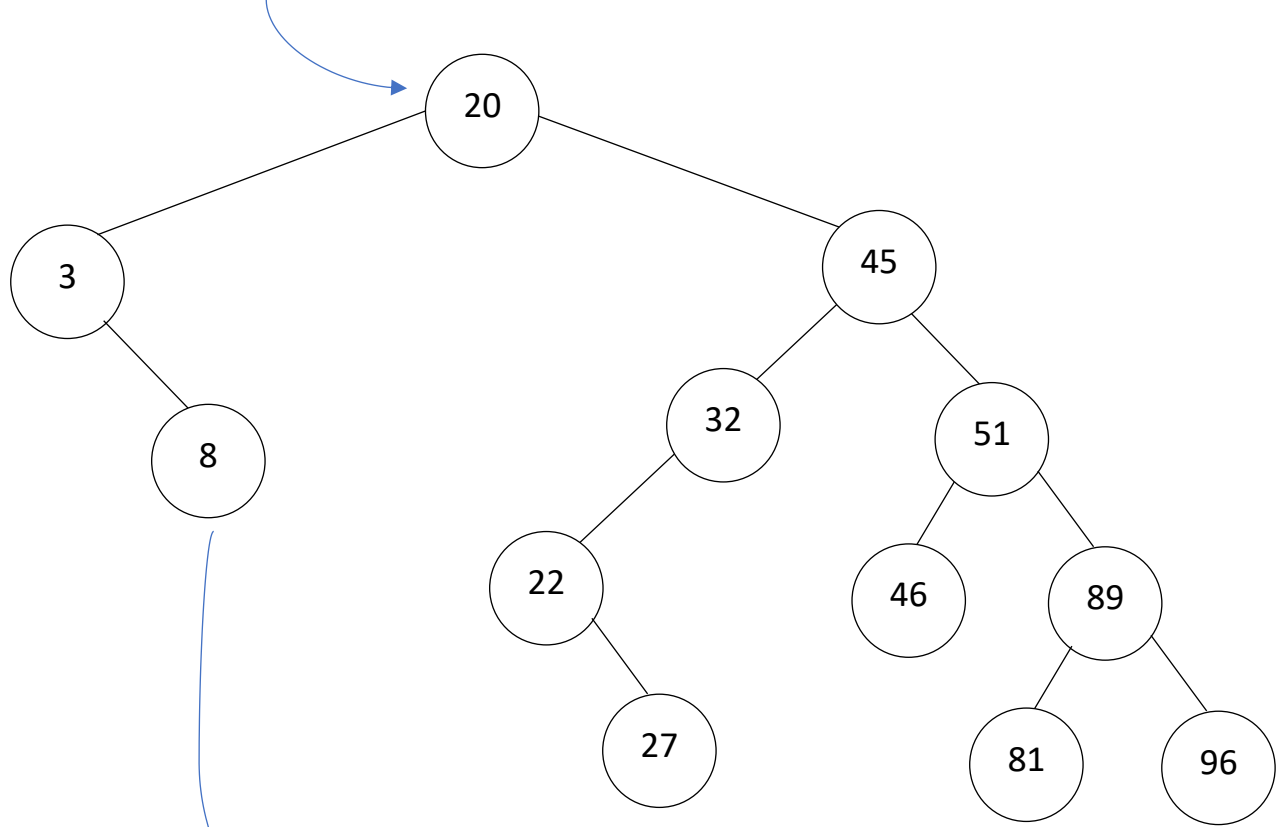
- 9) Simule a inserção dos seguintes valores em uma árvore binária: 20, 45, 32, 51, 89, 46, 81, 96, 22, 27, 3, 8, 9, 11, 12, 87, 55.

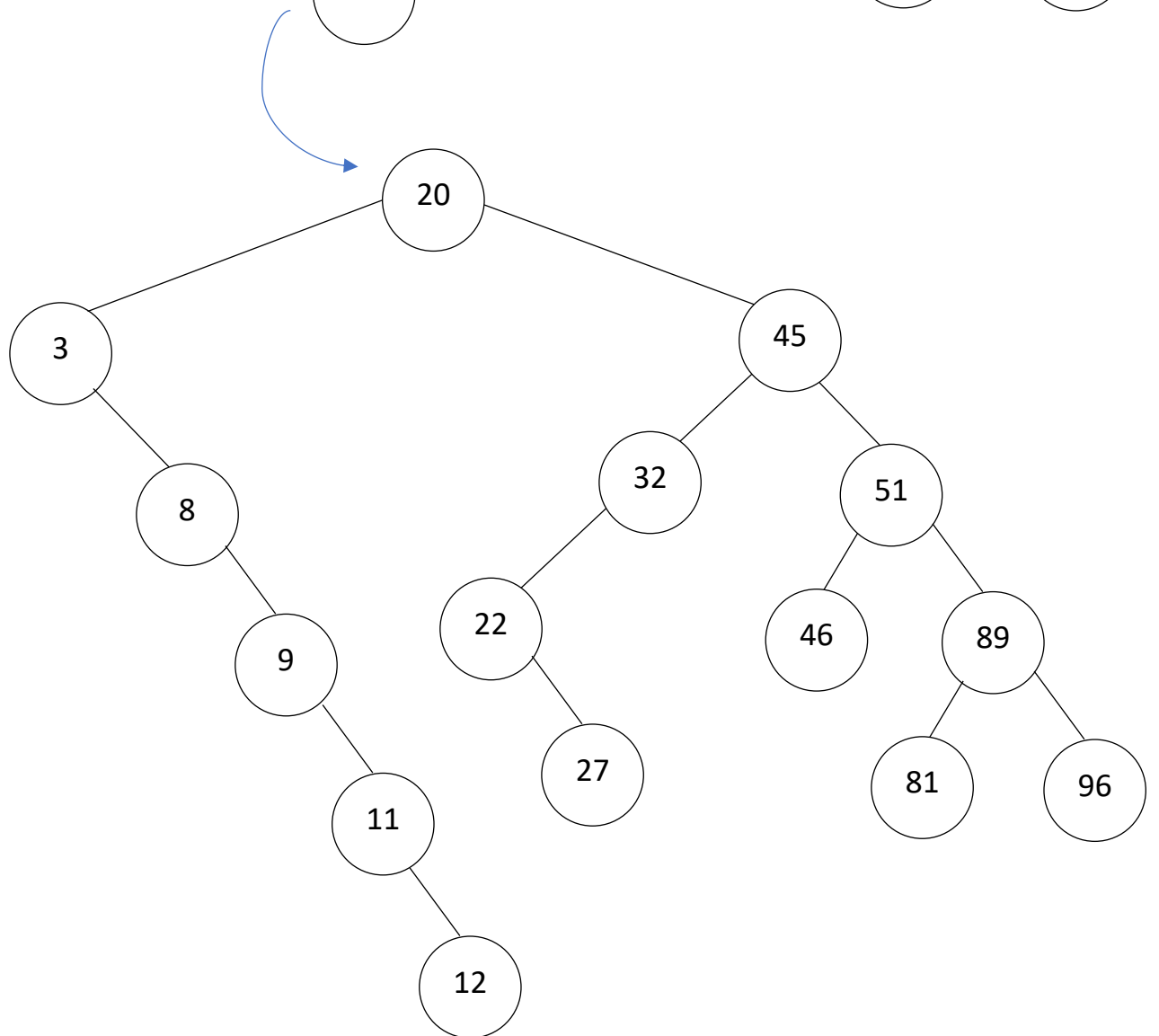
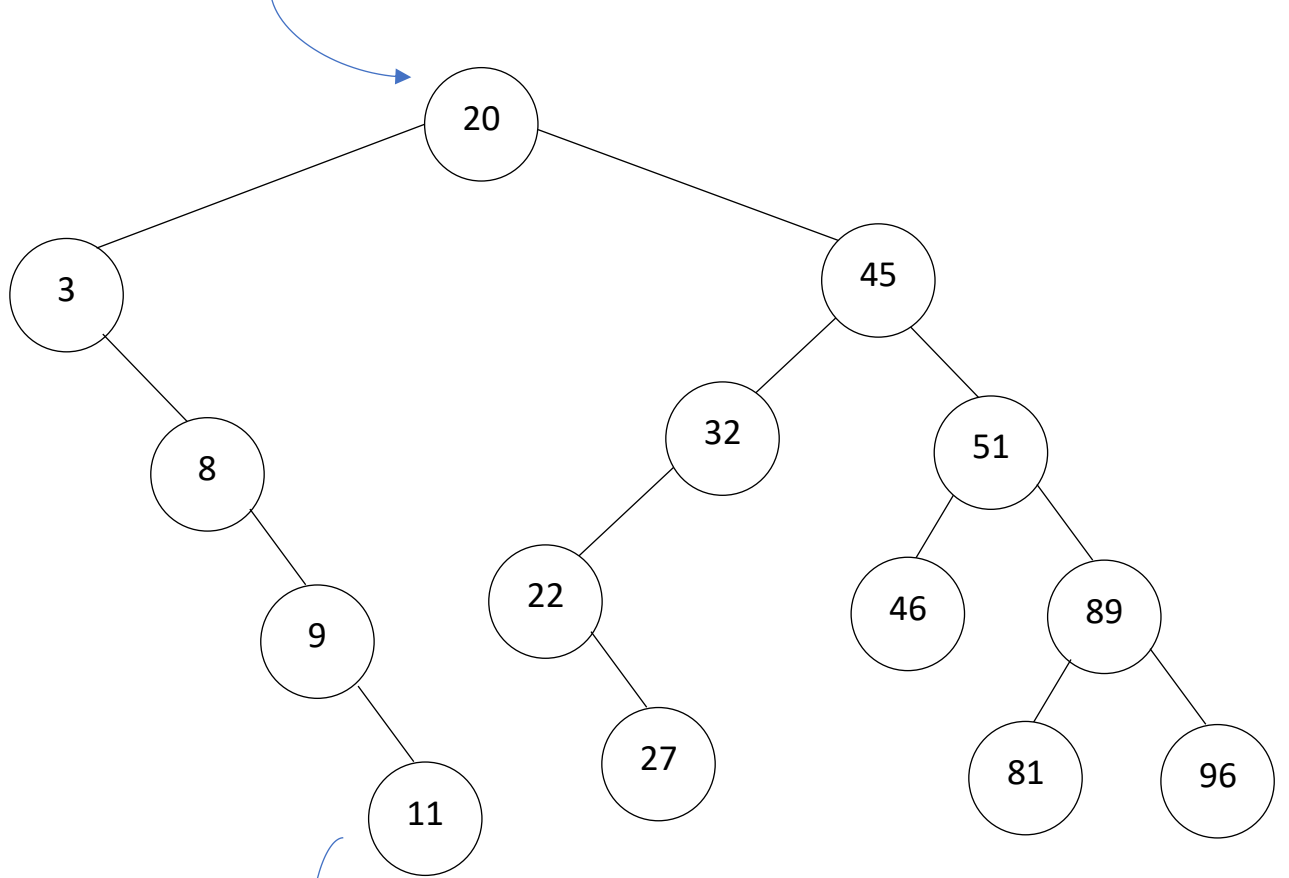


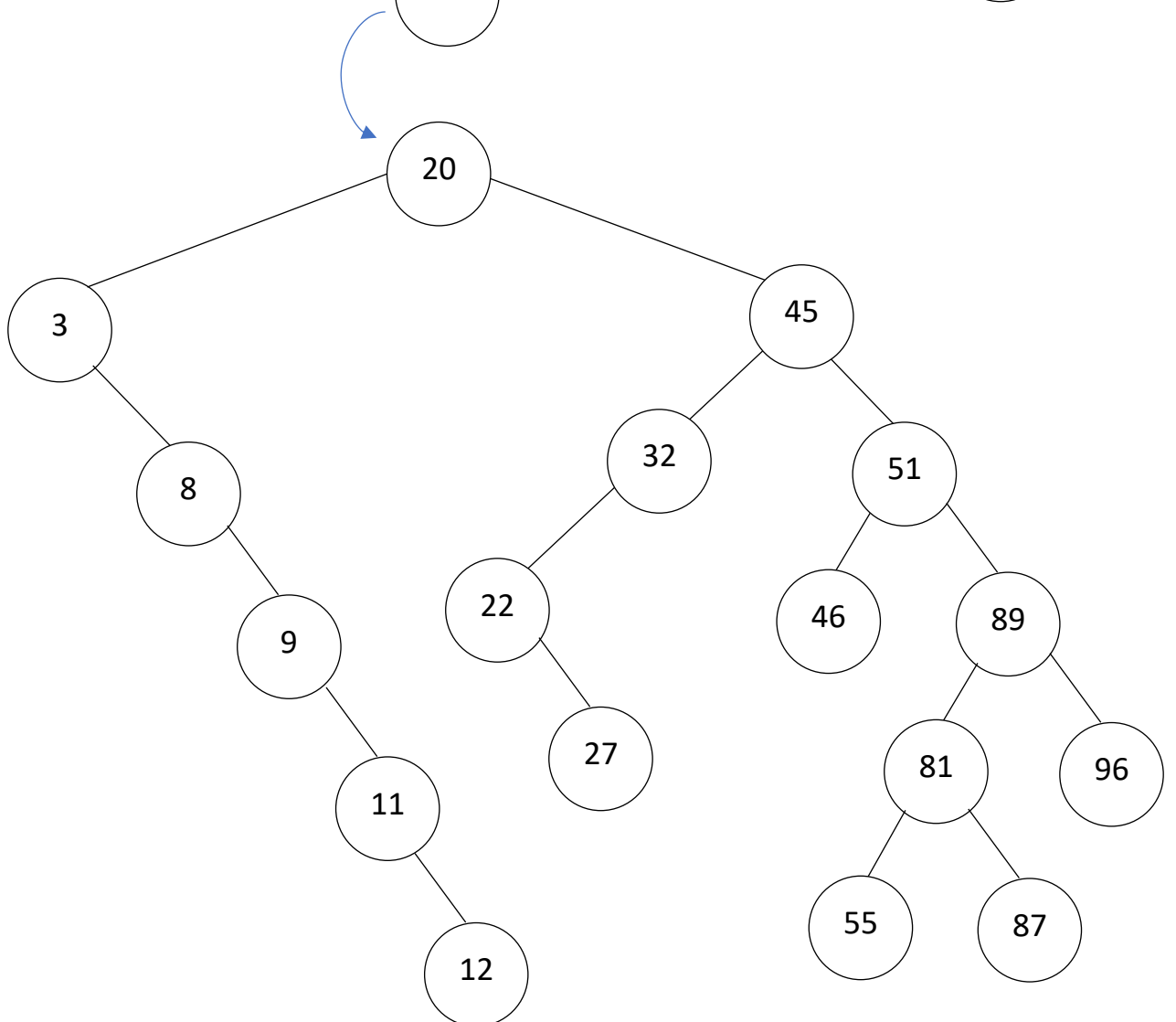
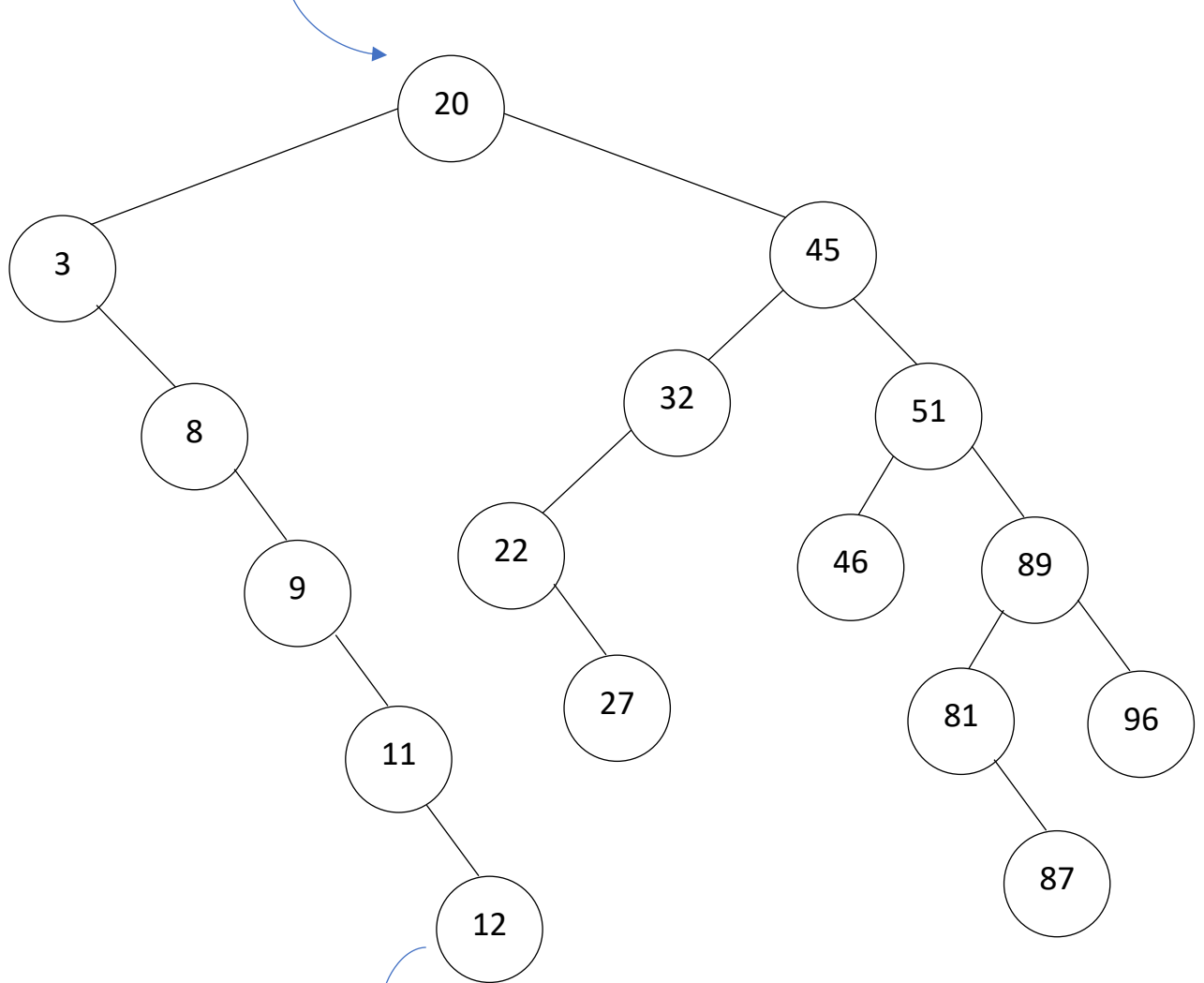




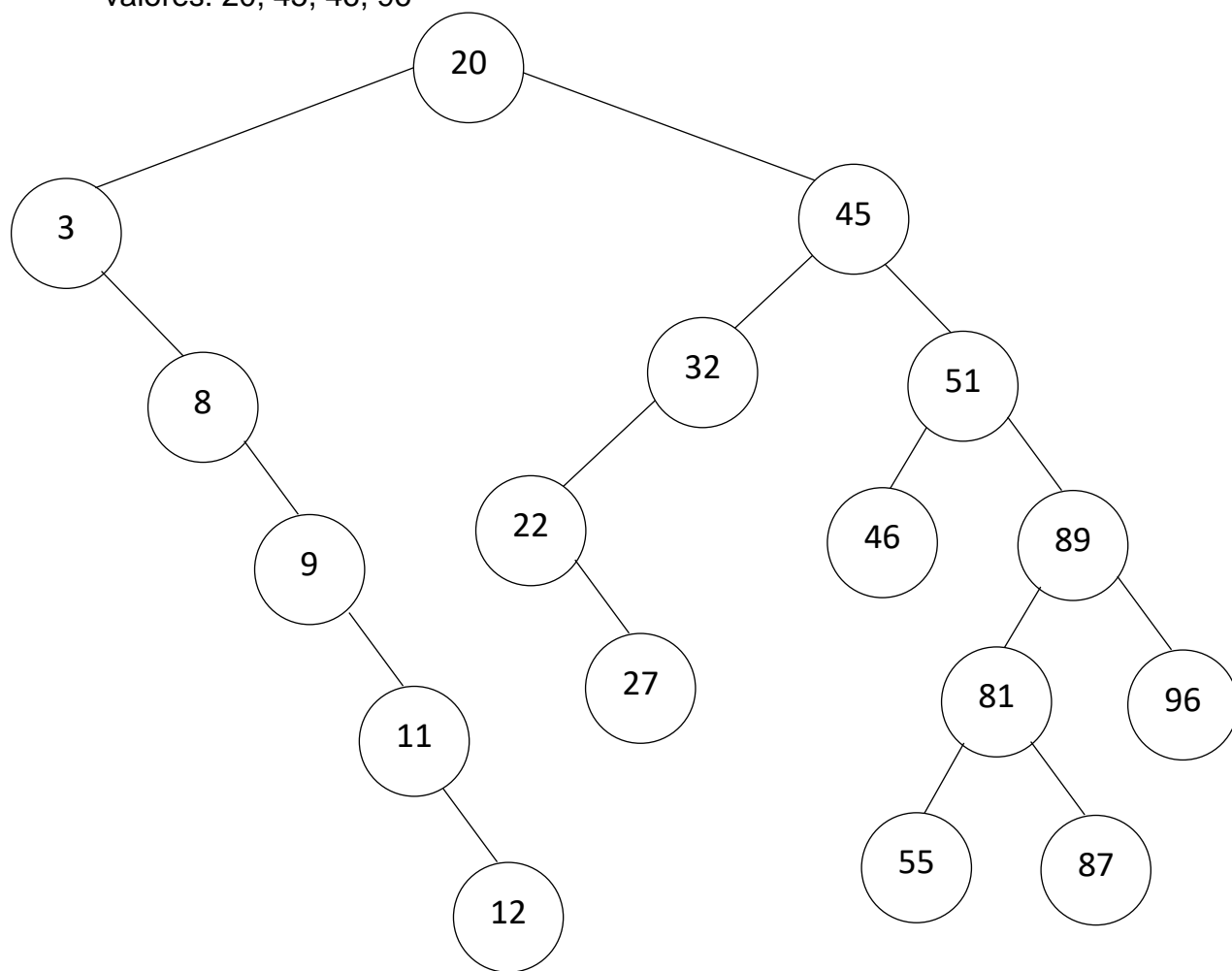




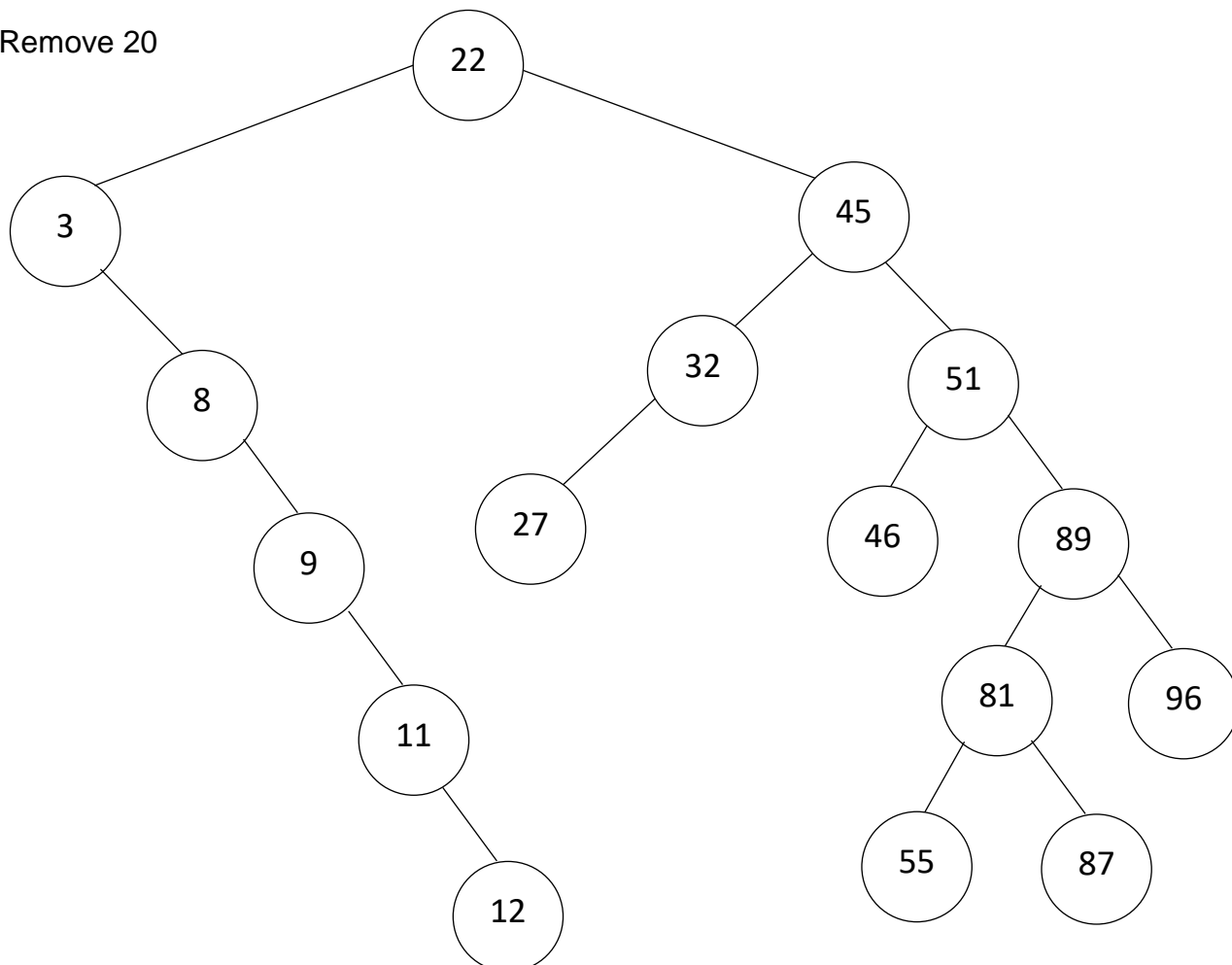




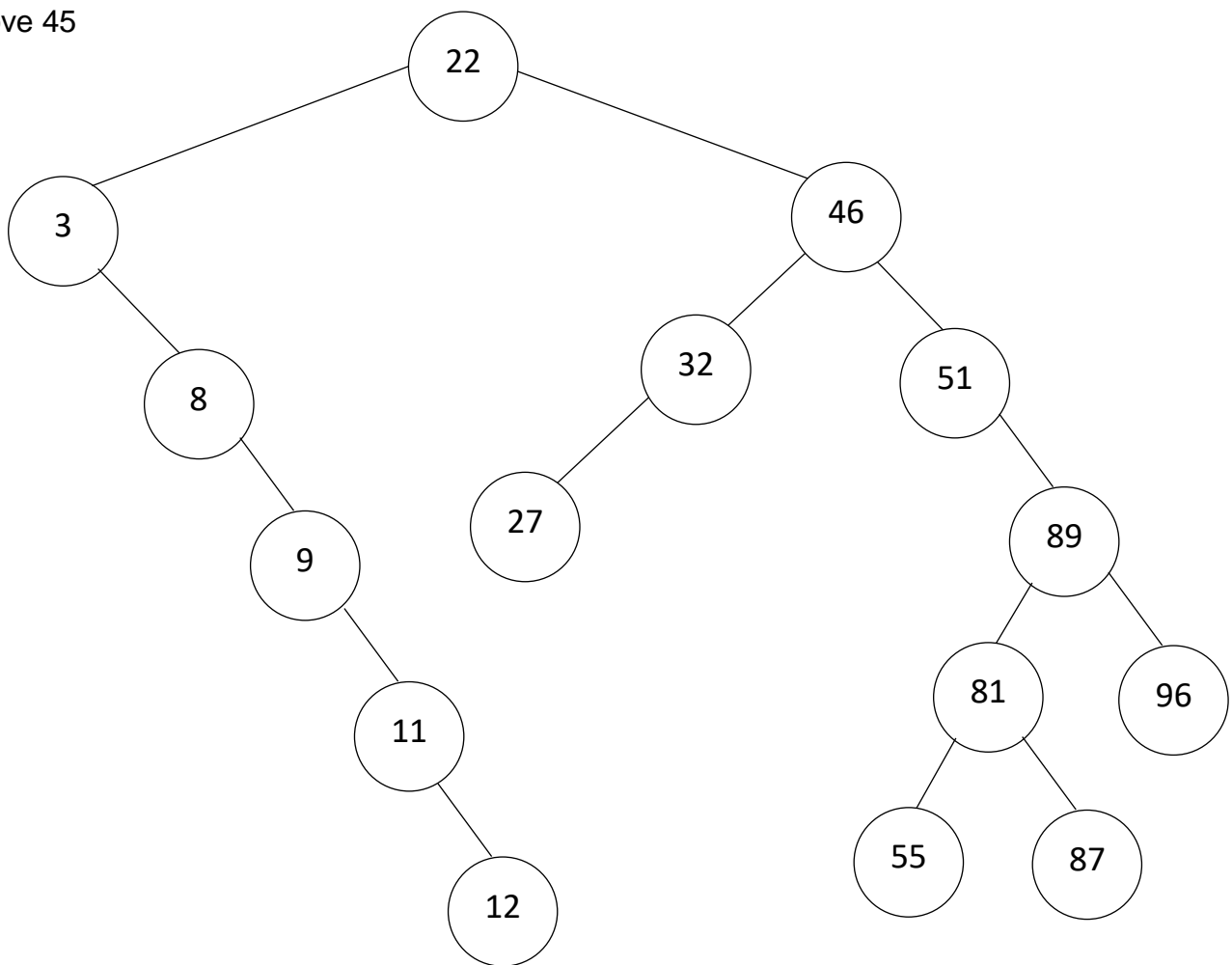
10) Considerando a árvore binária proposta no exercício 9, simule a exclusão dos seguintes valores: 20, 45, 46, 96



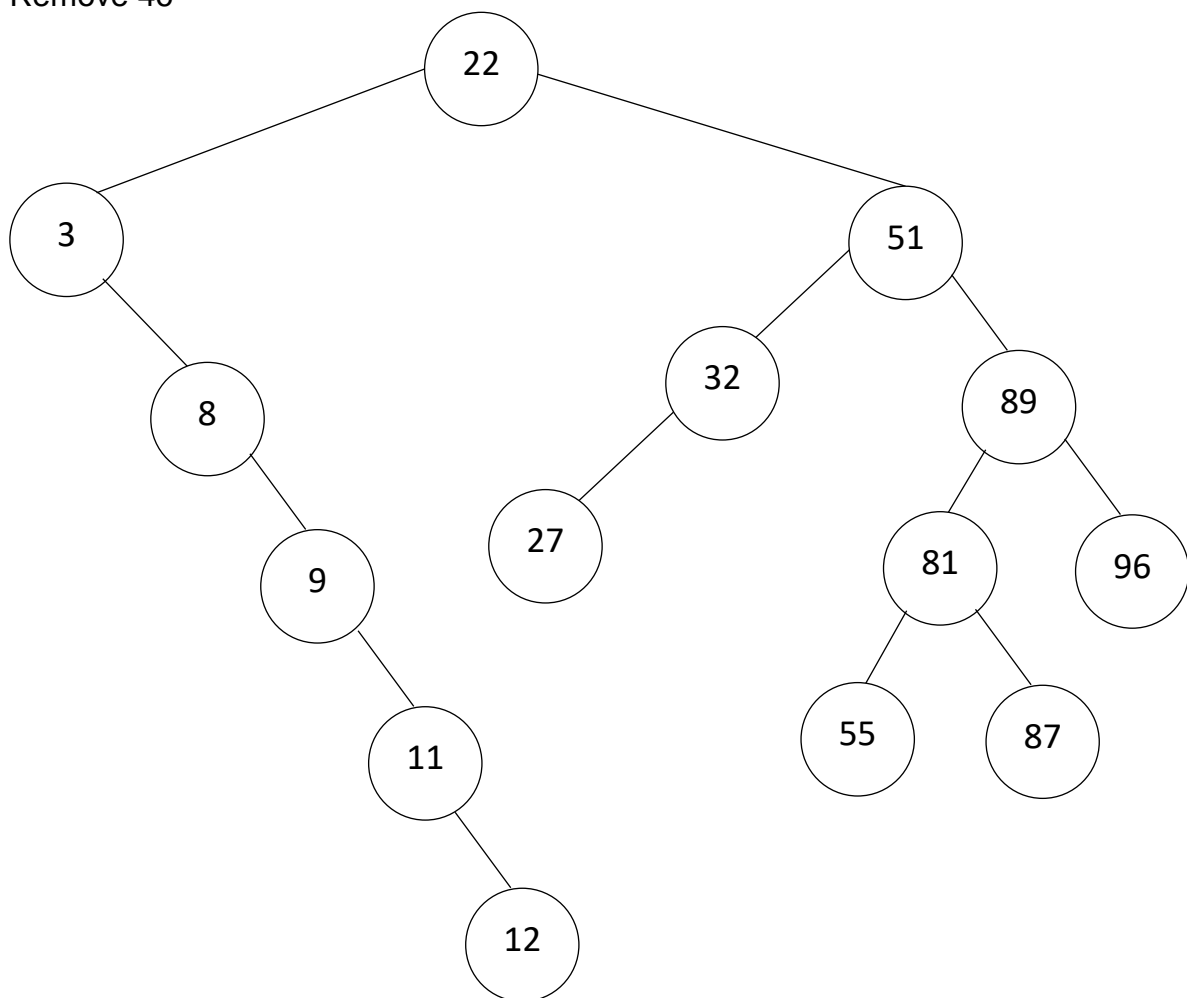
Remove 20



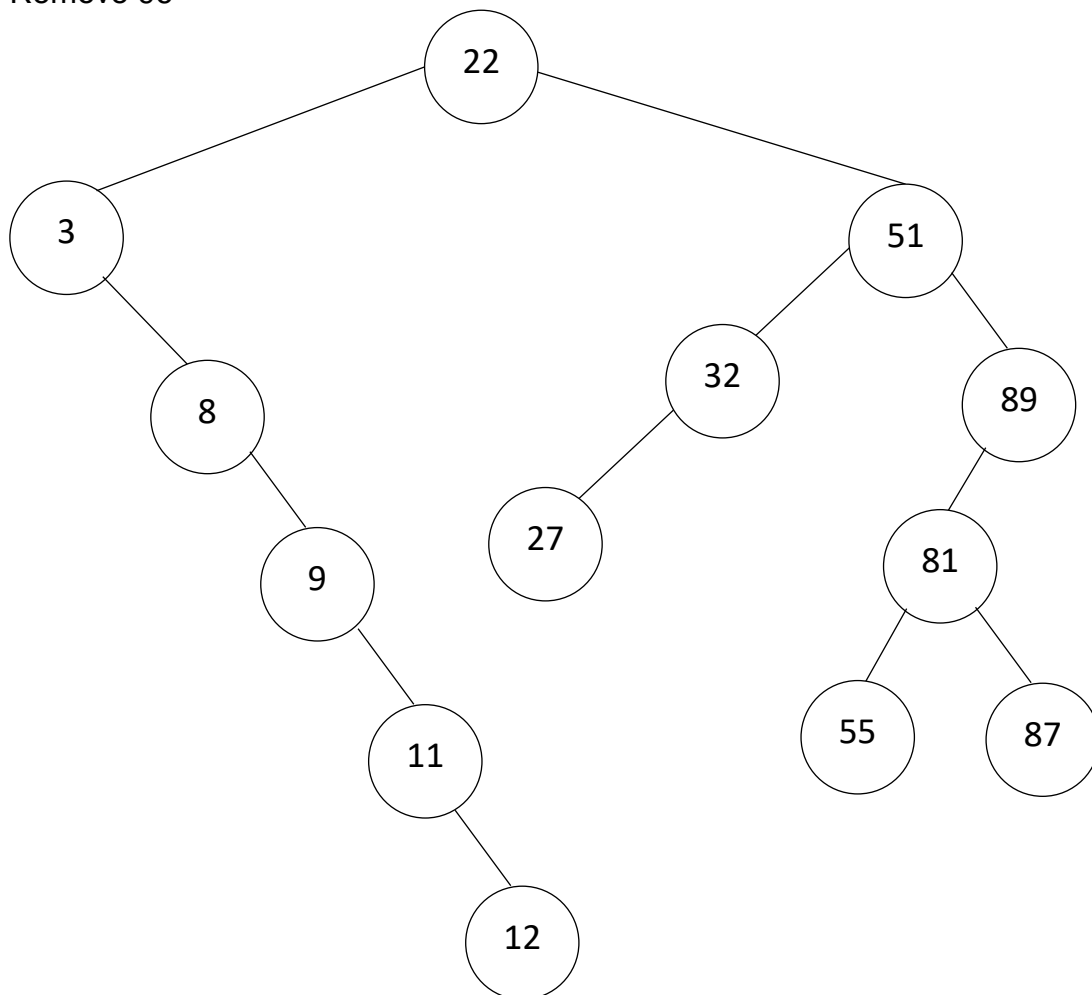
Remove 45



Remove 46



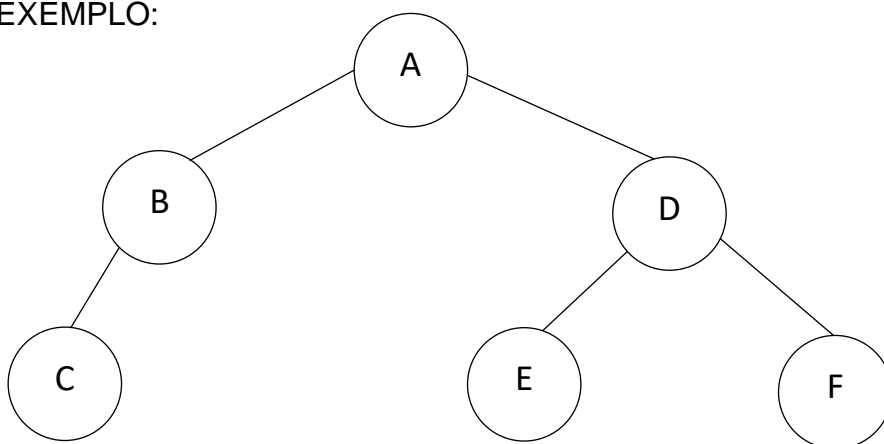
Remove 96



11) Apresente um exemplo de uso em árvores dos percurso em pré-ordem e pós-ordem.

Em algumas aplicações, é necessário percorrer uma árvore de forma sistemática, visitando cada nó da árvore uma única vez, em determinada ordem. Por exemplo, se cada nó da árvore possui um campo que armazena o salário, então podemos querer visitar cada nó para fazer um reajuste salarial. A visita seria atualizar o campo salário. Não podemos esquecer nenhum nó, nem queremos visitar um nó mais do que uma vez. Neste caso, a ordem de visita não é importante. Mas em algumas outras aplicações, queremos visitar os nós em certa ordem desejada.

EXEMPLO:



Pré-ordem: visite a raiz, então visite a sub-árvore da esquerda, depois a sub-árvore da direita

Pré-Ordem: [A, B, C, D, E, F]

Em-ordem: visite a sub-árvore da esquerda, então visite a raiz, depois a sub-árvore da direita

Em-Ordem: [C, B, A, E, D, F]

Pós-ordem: visite a sub-árvore da esquerda, então visite a sub-árvore da direita, depois a raiz

Pós-Ordem: [C, B, E, F, D, A]

12) Explique a diferença de uma árvore binária balanceada e de uma árvore binária não balanceada e quais estratégias podem ser utilizadas para balancear uma árvore binária.

Uma árvore binária balanceada é uma árvore binária na qual as alturas das duas sub-árvores de todo nó nunca difere em mais de 1. O balanceamento de um NÓ é definido como a altura de sua subárvore esquerda menos a altura de sua subárvore direita. Ou seja, para uma árvore ser considerada balanceada suas sub-árvores à esquerda e à direita devem possuir a mesma altura ou no máximo uma unidade. Se a altura dos nós filhos diferirem por mais de uma unidade, são chamadas de não balanceadas.

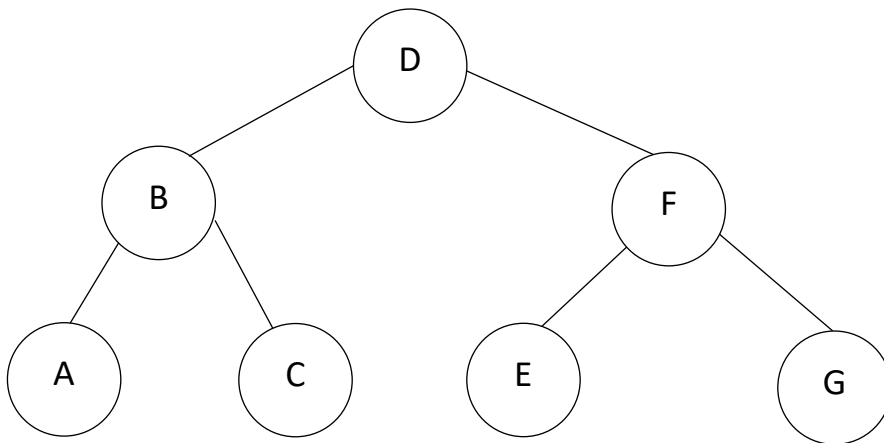
Para balancear uma árvore binária é necessário calcular o Fator de Balanceamento para verificar qual rotação deve ser efetuada a fim de rebalanceá-la.

$FB = h \text{ da subárvore direita} - h \text{ da subárvore esquerda}$

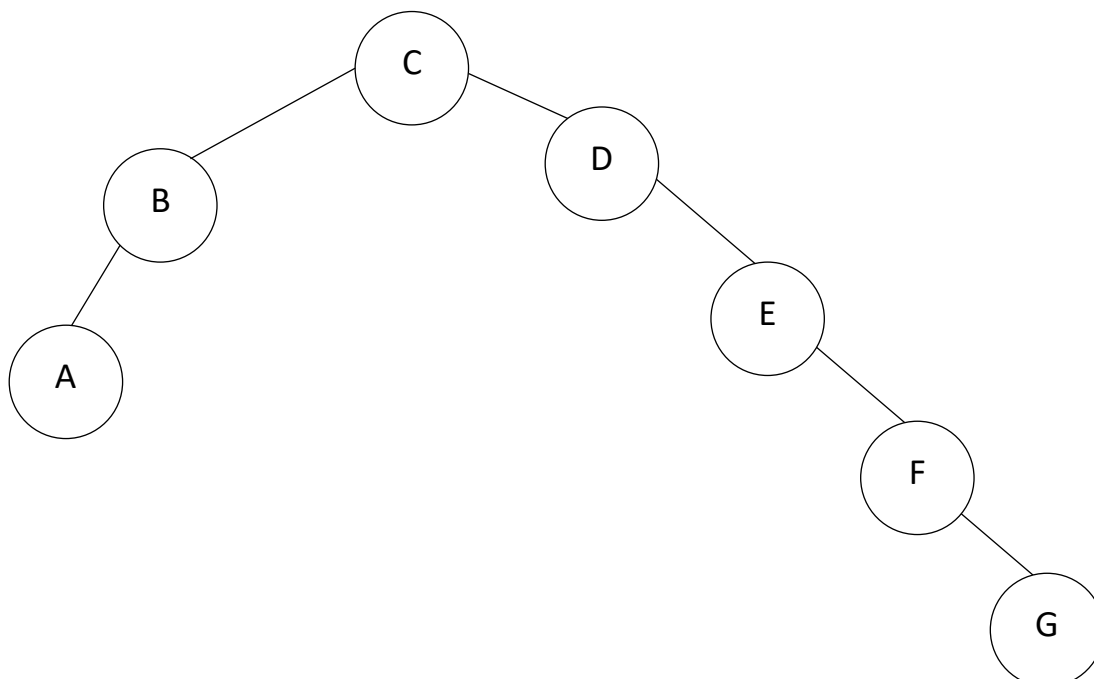
Se FB é negativo, as rotações são feitas à direita

Se FB é positivo, as rotações são feitas à esquerda

Árvore Balanceada:



Árvore Binária Não Balanceada:



13) Utilizando o conceito de grafos realize a implementação dos algoritmos de busca em largura e busca em profundidade e explique as suas principais diferenças.

A principal diferença é que a busca em largura utiliza uma fila para armazenar vértices que foram descobertos e precisam ser explorados, enquanto que a busca em profundidade utiliza uma pilha, fazendo com que a busca siga em profundidade.