

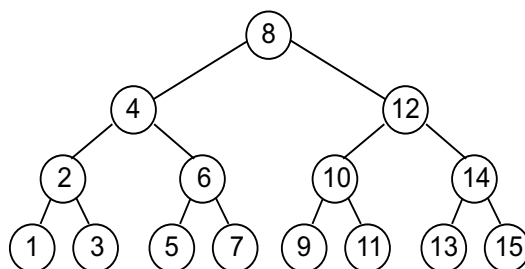
**Gabarito da Segunda Avaliação à Distância**

1. (1,0) Falso ou verdadeiro? (Justifique.) O fator de carga de qualquer tabela de dispersão é no máximo igual a 1.

Resposta: Falso. O fator de carga é dado pelo número de chaves dividido pelo número de compartimentos da tabela. Logo, se o número de chaves for maior que o de compartimentos, o fator de carga será maior que 1.

2. (1,5) Desenhe uma árvore binária de busca CHEIA COM ALTURA 4, colocando dentro de cada nó o valor de sua chave. As chaves são  $1, 2, \dots, k$  ( $k$  é o número de nós da árvore, que é um valor que você deve deduzir). A seguir, escreva a sequência de chaves que corresponde ao percurso em PRÉ-ORDEM desta árvore.

Resposta: O percurso em pré-ordem desta árvore é: 8 4 2 1 3 6 5 7 12 10 9 11 14 13 15.

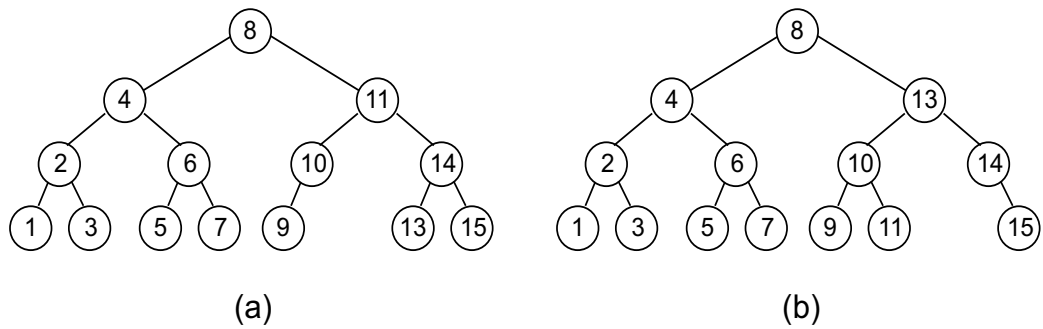


3. (1,5) Suponha que você deseja remover um nó de uma árvore binária de busca. Após removê-lo, como você deve reestruturar a árvore de modo que ela continue sendo uma árvore binária de busca? Dê um exemplo que mostre seu raciocínio. (Sugestão: use a árvore que você desenhou na questão anterior, removendo um nó qualquer e reestruturando-a.)

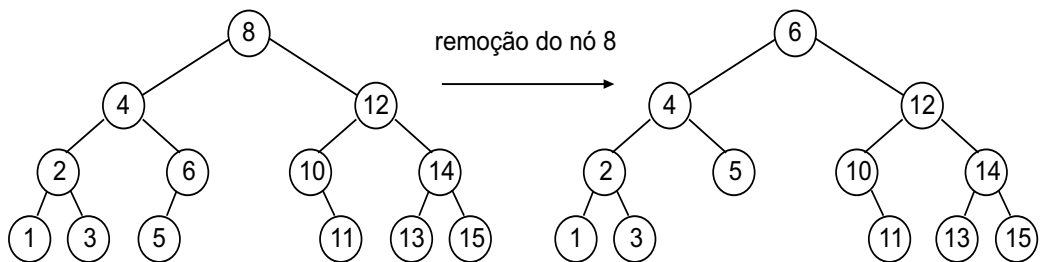
Resposta: Se o nó removido for uma folha, não há necessidade de reestruturação, pois a árvore resultante continuará sendo binária de busca. Em caso contrário (remoção de um nó interno), podemos:

- (a) substituir o nó removido pelo nó mais à direita (de maior valor) de sua subárvore esquerda (se este nó for uma folha), ou
- (b) substituir o nó removido pelo nó mais à esquerda (de menor valor) de sua subárvore direita (se este nó for uma folha).

No exemplo da árvore da questão anterior, ao removermos o nó 12, por exemplo, obteríamos uma das duas árvores abaixo, dependendo da estratégia que adotássemos ((a) ou (b)).

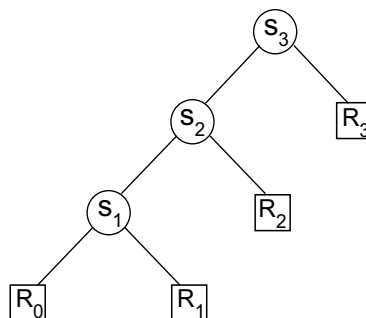


Caso a árvore binária de busca não seja cheia, e tanto o nó de maior valor da subárvore esquerda do nó removido quanto o de menor valor da subárvore direita não sejam folhas, então basta selecionar um destes nós para substituir o nó removido, e “promover” a subárvore deste nó para sua antiga posição, como no exemplo da figura a seguir.



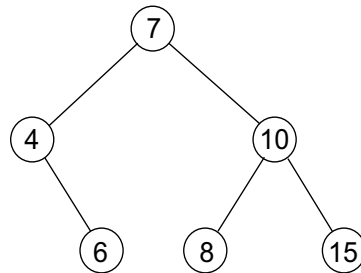
4. (1,5) Seja  $T$  uma árvore binária de custo mínimo relativa às frequências  $f_1, f_2, f_3, f'_0, f'_1, f'_2, f'_3$ . Escreva valores para estas frequências de modo que  $T$  seja uma árvore zigue-zague.

Resposta: Para as frequências  $f_1 = 1, f_2 = 2, f_3 = 4, f'_0 = f'_1 = f'_2 = f'_3 = 0$  temos a seguinte árvore binária de custo mínimo.



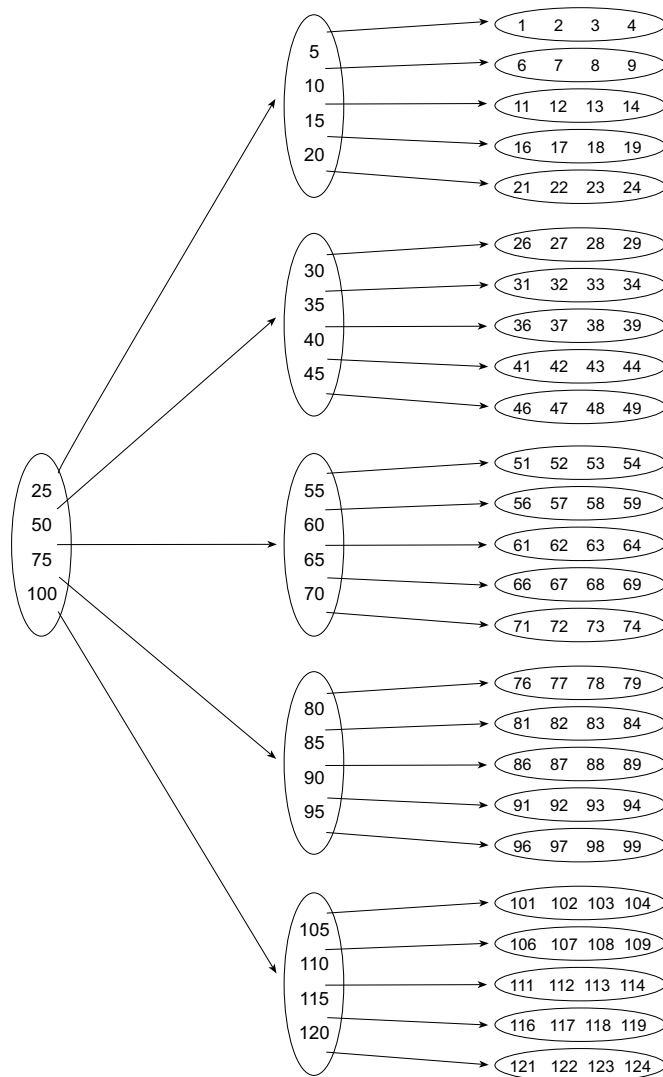
5. (1,5) A partir de uma árvore inicialmente vazia, desenhe a árvore AVL resultante da inserção dos nós com chaves 10,4,15,8,7,6 (nesta ordem.)

Resposta:



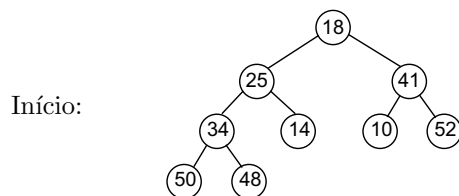
6. (1,5) Desenhe uma árvore B de ordem  $d = 2$  com três níveis e o maior número possível de chaves. (Os valores das chaves ficam à sua escolha.)

Resposta:

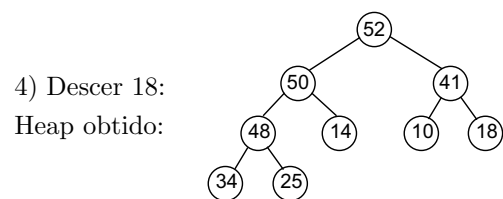
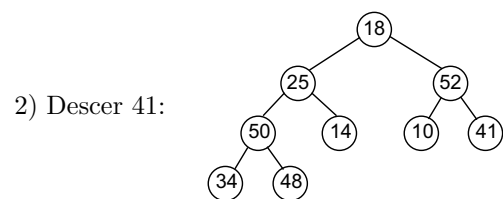
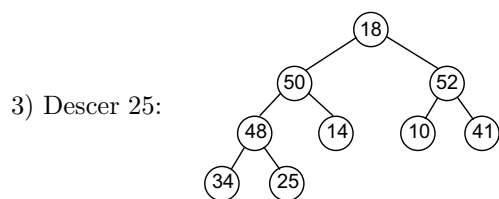
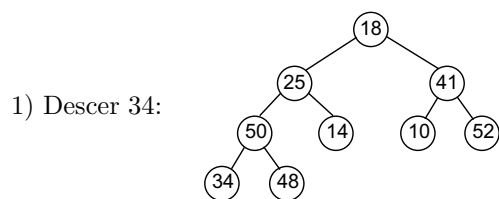


7. (1,5) Construa um heap com as seguintes prioridades: 18, 25, 41, 34, 14, 10, 52, 50, 48. A seguir, redesenhe o heap após a remoção de sua raiz.

Resposta:



Construção do heap:



Após a remoção da raiz:

