Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP2 - Segundo Semestre de 2015

Nome -
Assinatura -

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

---

1. (2,0) Desenhe uma árvore T_1 com as seguintes características:
   (a) T_1 é uma árvore binária de busca;
   (b) T_1 tem altura 4;
   (c) T_1 é estritamente binária;
   (d) T_1 tem o menor número possível de nós;
   (e) os nós de T_1 contêm as chaves $1, 2, \ldots, n$, onde n é o número de nós.

   Não se esqueça de colocar um valor de chave dentro de cada nó.

   Resposta: Uma árvore estritamente binária possui o menor número de nós quando em cada nível existe exatamente um nó com dois filhos, a menos do último nível. A Figura 1 mostra um exemplo de tal árvore.

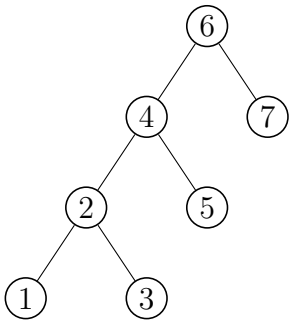
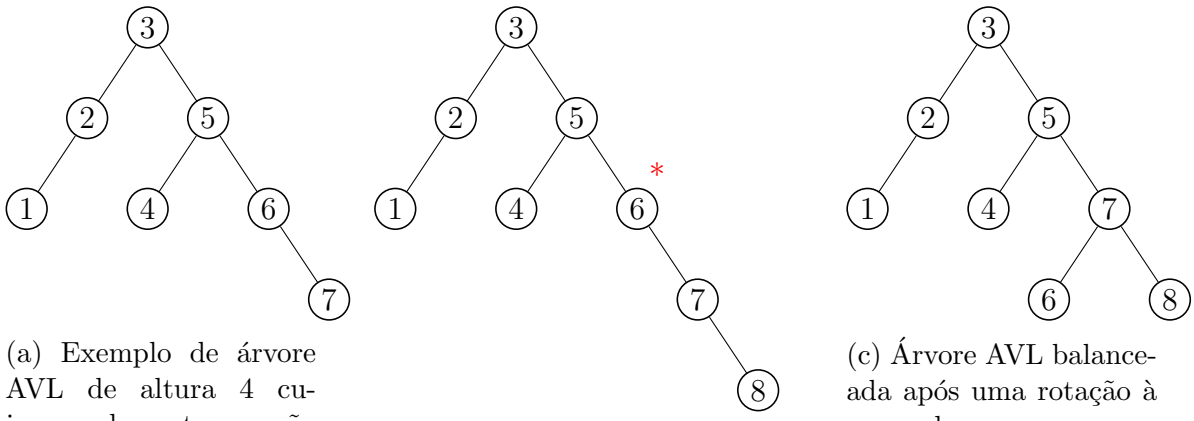


Figura 1: Representação de uma árvore de busca estritamente binária com altura 4 e número mínimo de nós.

2. (2,0) Desenhe uma árvore T_2 com as seguintes características:
   (a) T_2 é uma árvore AVL;
   (b) T_2 tem altura 4;
   (c) a inserção em T_2 de um novo nó com chave de valor *maior* do que todos os valores presentes em T_2 exige a realização de uma *rotação esquerda*.

   Não se esqueça de colocar um valor de chave dentro de cada nó.

   Resposta: A Figura 2a mostra um exemplo de tal árvore com os elementos de 1 a 7. A Figura 2b mostra a árvore obtida após a inclusão do elemento 8, onde o nó que contém o elemento 6 passa a ser desbalanceado. A Figura 2c mostra a árvore resultante após uma rotação à esquerda para torná-la balanceada.



(a) Exemplo de árvore AVL de altura 4 cujos elementos são $1, 2, 3, 4, 5, 6, 7$.

(b) Inclusão do elemento 8.

(c) Árvore AVL balanceada após uma rotação à esquerda.

Figura 2: Exemplo de inclusão de um elemento maior que todos os pertencentes a uma árvore AVL de altura 4.

3. (2,0) Desenhe uma árvore T_3 com as seguintes características:
   (a) T_3 é uma árvore B;
   (b) T_3 tem ordem $d = 2$;
   (c) T_3 tem altura $h = 3$;
   (d) A remoção da chave de *menor valor* dentre todas as chaves presentes em T_3 exige a realização de uma *concatenação* de páginas.

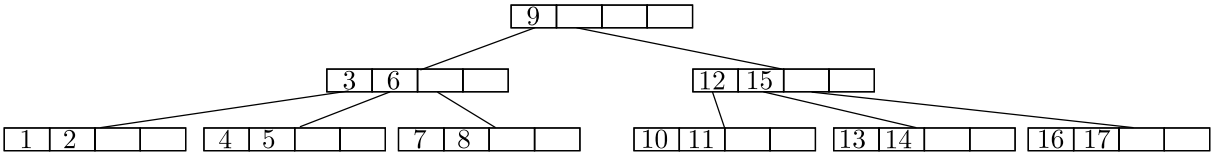
   Não se esqueça de colocar valores de chaves dentro de cada página.

   Resposta: A Figura 3a mostra um exemplo de árvore B cuja remoção do menor elemento exige uma concatenação de páginas.

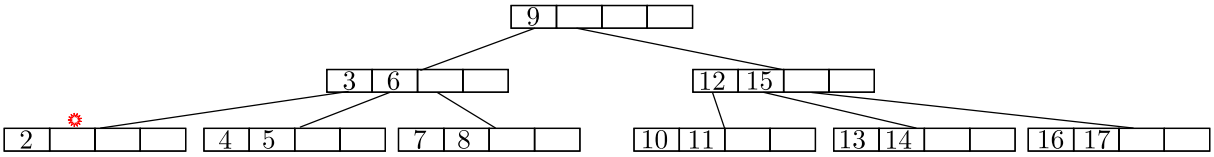
4. (2,0) **[Questão anulada.]** Desenhe uma árvore T_4 com as seguintes características:
   (a) T_4 é um *heap* (lista de prioridades);
   (b) T_4 tem $n = 10$ nós;
   (c) as prioridades dos nós são $1, 2, \ldots, n$;
   (c) o nó com prioridade 4 é o filho esquerdo da raiz de T_4.

   Não se esqueça de colocar um valor de prioridade dentro de cada nó.

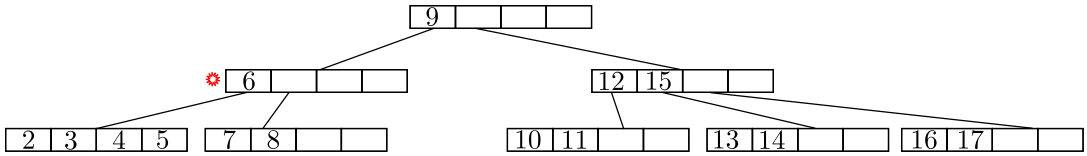
   Resposta: Note que o heap desejado não pode ser de prioridade máxima. Isso se deve porque o elemento 4 deve ser filho esquerdo da raiz de T_4, uma



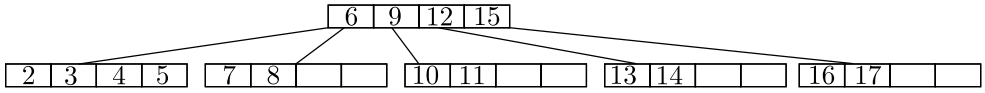
(a) Árvore B inicial.



(b) A página marcada possui menos que d elementos após a remoção do elemento 1.



(c) Árvore obtida após a concatenação.



(d) Árvore final.

Figura 3: Representação da remoção do elemento de menor valor da árvore.

vez que existem apenas 3 elementos menores que 4, mas as sub-árvores do nó de elemento 4 devem possuir 5 elementos no total. Por este motivo a questão está **anulada**.

Porém, poderíamos construir um *heap de prioridade mínima* para este caso. Nestes heaps a prioridade em cada nó é tal que ela é menor ou igual que as prioridades das raízes de suas sub-árvores. Além disso, cada sub-árvore representa um heap de prioridade mínima. Portanto T_4 pode ser descrito como um heap de prioridade mínima como na Figura 4.

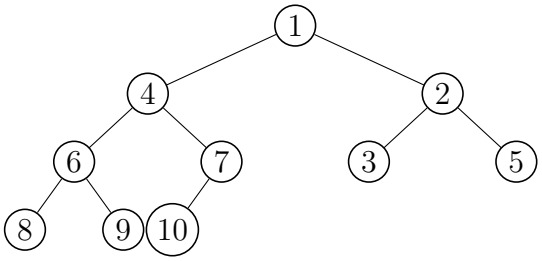


Figura 4: Árvore que representa um heap com elementos de 1–10 e cujo elemento esquerdo da raiz possui valor 4.

5. (2,0) Deseja-se construir uma árvore de Huffman T_5 para as seguintes frequências: f_1, f_2, f_3, f_4, f_5 e f_6. Forneça valores para estas frequências de modo que, ao executar o algoritmo guloso para a construção de T_5, *nunca* ocorra a possibilidade de diferentes escolhas para a fusão de duas sub-árvores, em qualquer iteração.

   Resposta: Para evitar a possibilidade de fusões distintas de sub-árvores, devemos forçar que todas as sub-árvores geradas em cada passo do algoritmo possuam valores distintos em suas raízes. Uma forma simples de fazer isso é escolher valores de frequências satisfazendo a seguinte propriedade:

$$\sum_{i=1}^{k-1} f_i < f_k, \text{ para todo } 1 \leq k \leq 6.$$

   A Figura 5 mostra um exemplo satisfazendo essa propriedade.

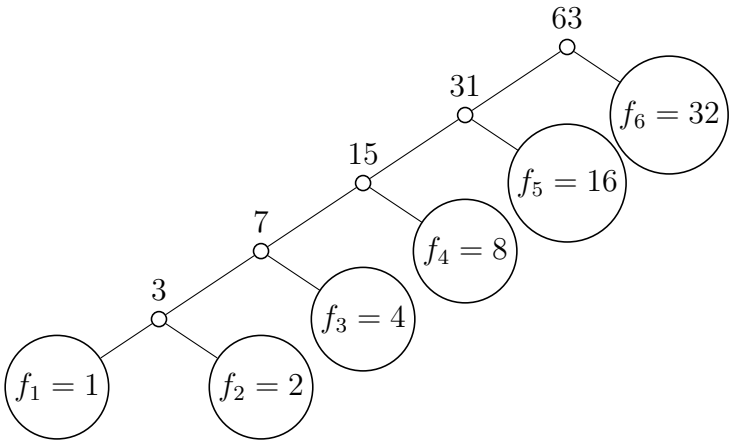


Figura 5: Exemplo de árvore de Huffman ótima onde em cada passo existe apenas uma escolha de árvores a serem escolhidas.