Estrutura de Dados - 1o. período de 2013

Gabarito da Segunda Avaliação à Distância

Todas as questões valem 1,0 ponto.

1. Determinar a árvore binária de busca ótima relativa às seguintes frequências: $f_1 = 1, f_2 = 1, f_3 = 1, f'_0 = 0, f'_1 = 0, f'_2 = 0, f'_3 = 0$. Determine as matrizes F, c e k associadas ao algoritmo, que estão descritas no livro-texto.

   Resposta: As matrizes do algoritmo de cálculo da árvore ótima são:

   Matriz dos custos $c[i, j]$:

0	1	3	5
-	0	1	3
-	-	0	1
-	-	-	0

   Matriz dos valores $F[i, j]$:

0	1	2	3
-	0	1	2
-	-	0	1
-	-	-	0

   Matriz dos valores minimizantes k:

-	1	1 (2)	2
-	-	2	2 (3)
-	-	-	3
-	-	-	-

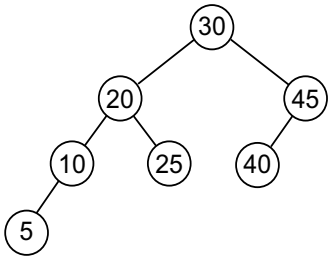
   Da última matriz acima obtemos a árvore ótima de raiz s_2, filho esquerdo s_1 e direito s_3.

2. Ainda com relação às árvores binárias de busca ótimas, determine condições suficientes sobre as frequências $f_1, \ldots, f_n$ e $f'_0, f'_1, \ldots, f'_n$, para que a árvore ótima resultante tenha a seguinte característica: s_{i+1} é filho direito de s_i, para $i = 1, \ldots, n-1$.

   Resposta: $f_n = 1$, $f_{n-1} = 2$, $f_i = f_{i+1} + f_{i+2} + 1$, $1 \leq i \leq n-2$, e $f'_0 = f'_1 = \cdots = f'_n = 0$.

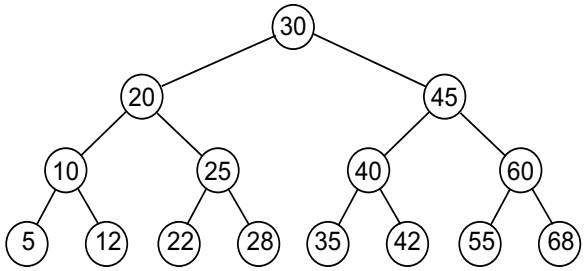
3. Desenhe uma árvore AVL de altura 4 que seja *minimal*, isto é, tal que a exclusão de qualquer de seus nós provoca a diminuição de sua altura.

   Resposta:



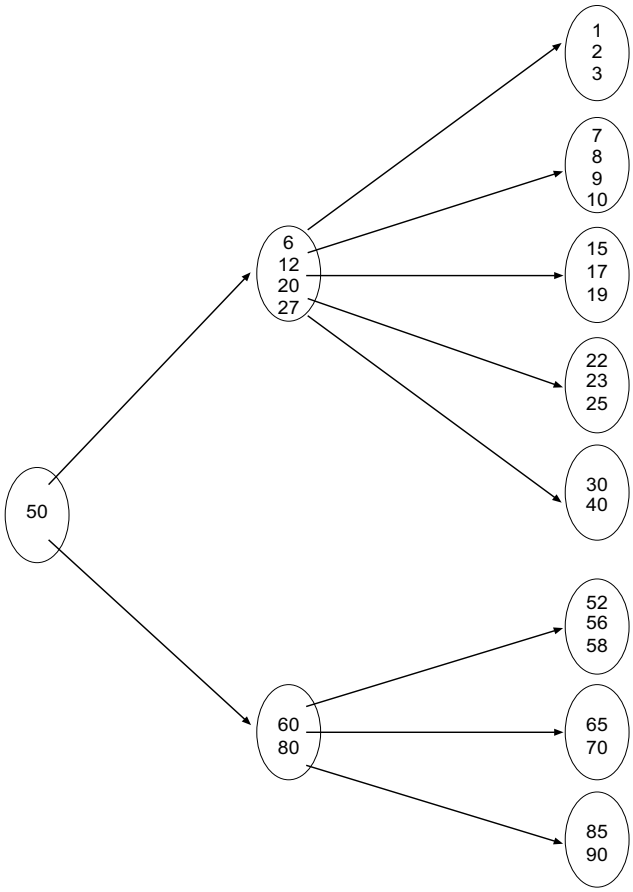
4. Desenhe uma árvore AVL de altura 4 que seja *maximal*, isto é, tal que a inclusão de qualquer nó provoca o aumento de sua altura.

   Resposta:

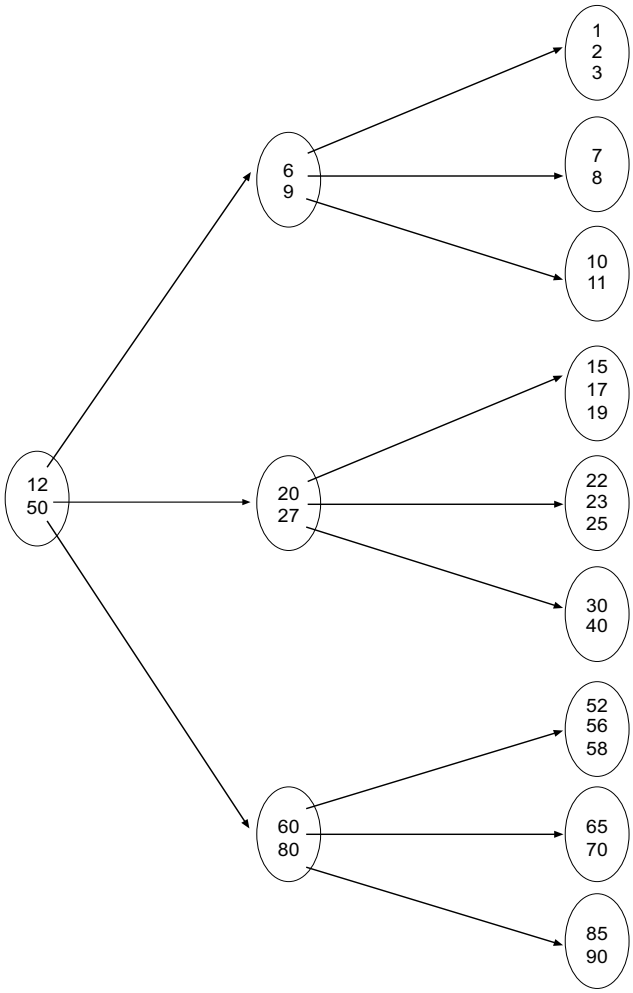


5. Desenhe uma árvore B de ordem $d = 2$ com três níveis. (Os valores nos nós ficam à sua escolha.) A seguir, escolha uma nova chave de forma que a sua *inserção* exija uma cisão propagável. Desenhe a árvore B resultante após a inserção.

   Resposta:



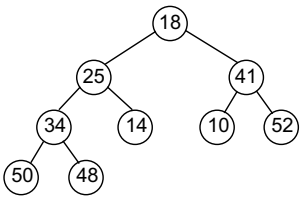
Inserindo a chave 11, temos uma cisão propagável, que resulta na seguinte árvore B:



6. Execute o método de ordenação por heap ("heapsort"), aplicando-o às seguintes prioridades (nesta ordem): $18, 25, 41, 34, 14, 10, 52, 50, 48$. Desenhe as configurações sucessivas da árvore durante o processo de ordenação.

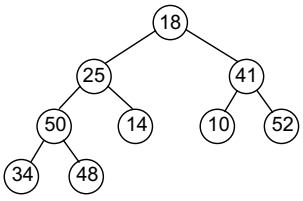
   Resposta:

   Início:

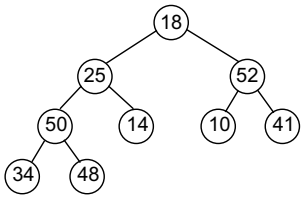


   Construção do heap: (comando *arranjar*(n))

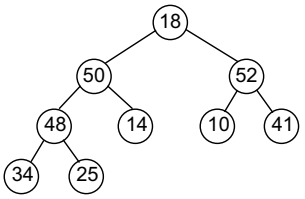
   Descer 34:



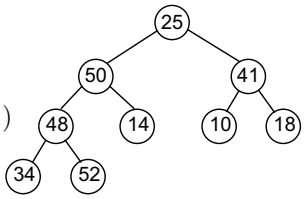
   Descer 41:



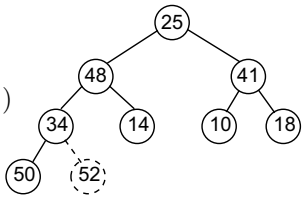
Descer 25:



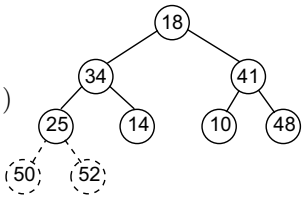
$m := 9$
trocar(TB[1],TB[9])



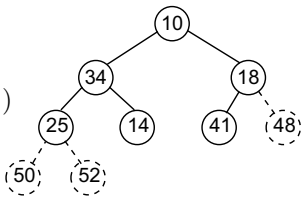
trocar(TB[1],TB[8])



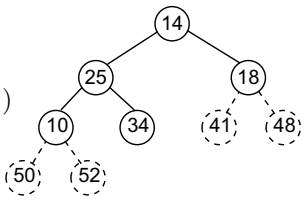
trocar(TB[1],TB[7])



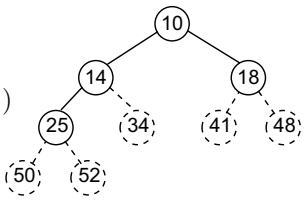
trocar(TB[1],TB[6])



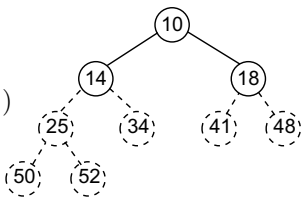
trocar(TB[1],TB[5])



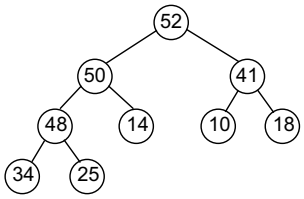
trocar(TB[1],TB[4])



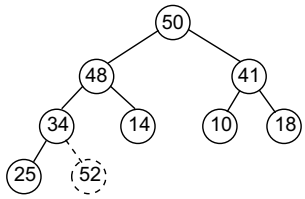
trocar(TB[1],TB[3])



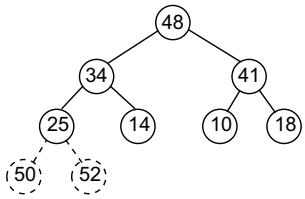
Descer 18:
Heap
obtido:



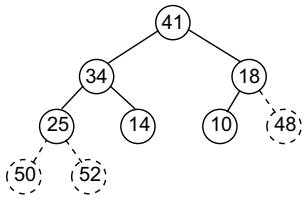
$m := 8$
descer(1,8)



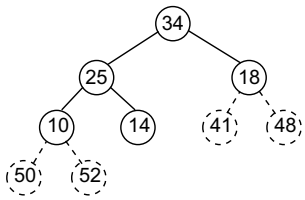
$m := 7$
descer(1,7)



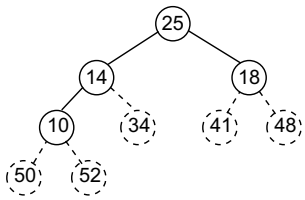
$m := 6$
descer(1,6)



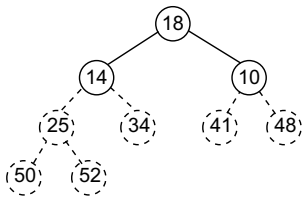
$m := 5$
descer(1,5)



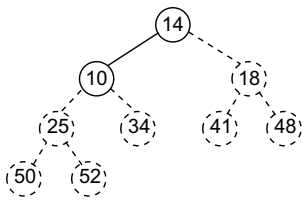
$m := 4$
descer(1,4)



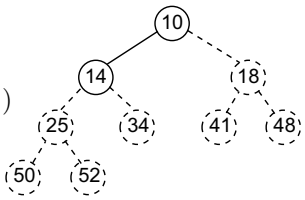
$m := 3$
descer(1,3)



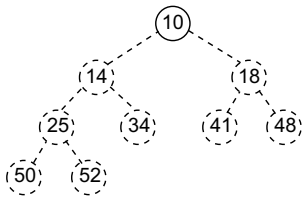
$m := 2$
descer(1,2)



trocar(TB[1],TB[2])

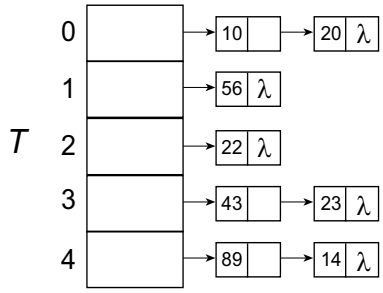


$m := 1$
descer(1,1)



7. Seja T uma tabela de dispersão com 5 posições implementada por encadeamento exterior. A função de dispersão é $h(x) = x \bmod 5$. Desenhe a tabela após a inclusão das chaves 43,89,56,23,14,22,10,20.

   Resposta:



8. Escreva um algoritmo que, dadas duas cadeias de caracteres X e Y, verifica se a cadeia X é prefixo ou sufixo ou ambas as coisas da cadeia Y. Exemplo: se $X = aba$ e $Y = abacataba$, então X é prefixo e sufixo de Y simultaneamente; ao passo que se $X = taba$, então neste caso X é apenas sufixo de Y.

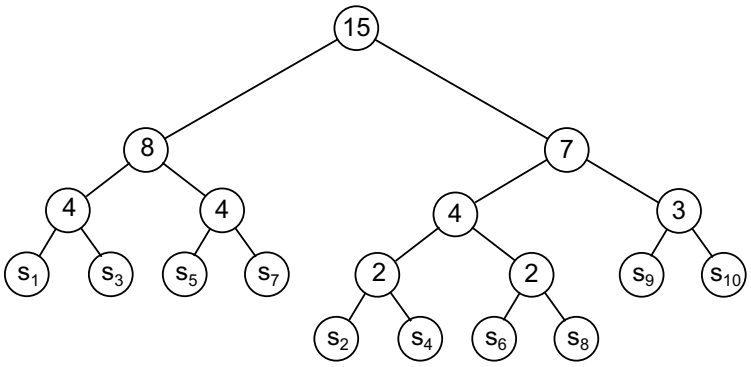
   Resposta: Sejam m o comprimento de X e n o comprimento de Y.

```
i := 1
pre := V
suf := V
enquanto i ≤ m faça
        se X[i] = Y[i] então
                i := i + 1
        senão
                i := m + 1
                pre := F
se pre = V então   imprimir ("X é prefixo de Y")
senão   imprimir ("X não é prefixo de Y")
i := 1
j := n − m + 1
enquanto j ≤ n faça
        se X[i] = Y[j] então
                i := i + 1
                j := j + 1
        senão
```

$j := n + 1$
$suf := F$
se $suf = V$ então *imprimir* ("X é sufixo de Y")
senão *imprimir* ("X não é sufixo de Y")

9. Construa uma árvore de Huffman para as frequências satisfazendo: $f_i = 1$ se i é par, $f_i = 2$ se i é ímpar. O índice i varia de 1 a 10.

   Resposta:



10. Responda: como é a árvore de Huffman relativa a n frequências iguais? (Suponha que n é da forma $n = 2^k$, isto é, n é uma potência de 2.)

    Resposta: É uma árvore cheia de altura $k + 1$. Inicialmente, temos 2^k árvores com um único nó (altura 1), de mesmo valor. O algoritmo vai unindo estas árvores duas a duas, até que todas façam parte de uma árvore cheia com 3 nós (altura 2). Como n é potência de 2, o algoritmo vai sempre unindo sempre duas árvores cheias em uma única árvore cheia, e somente une duas árvores de altura x quando não houver mais nenhuma árvore de altura menor que x. Ao final do algoritmo, duas árvores cheias de altura k são unidas em uma árvore cheia de altura $k + 1$.