



Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP3 - Segundo Semestre de 2016

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) É dado um vetor de caracteres V com $n > 0$ elementos, onde cada posição contém um caractere **e** ou **s**. É dada também uma fila **F** com $m > 0$ posições, que se encontra inicialmente vazia. O vetor **V** indica, da esquerda para a direita, o movimento de clientes na fila **F**. Cada caractere **e** indica a saída de um cliente da fila para ser atendido. Você deve desenhar o estado da fila **F** após a análise do vetor **V**. Exemplo: se $V = [e, e, e, s, e, s]$ e $m=4$, sua resposta será $F = [- - 3 4]$, isto é, o terceiro e o quarto clientes a chegar ainda estão aguardando na fila. Cada traço indica uma posição vazia no vetor que implementa a fila **F**.

Desenhe o estado da fila **F** nos seguintes casos:

(a.) (1,0) $V = [e, e, e, s, e, s, e]$ e $m = 5$.

R. $F = [- - 3 4 5]$

(b.) (1,0) $V = [e, e, e, s, e, s, e]$ e $m = 4$.

R. $F = [5 - 3 4]$

2. (2,0) Escreva um algoritmo que execute a seguinte tarefa: Dada uma lista não ordenada com n elementos, encontre o maior e o menor elementos desta lista. (A lista pode ser implementada de forma sequencial ou encadeada, fica à sua escolha.)

R. **Proc_encontrar**(L)

$menor := +\infty$

$maior := -\infty$

para $i := 1$ até $i \leq n$ faça

 se $L[i] > maior$ então

$maior := L[i]$

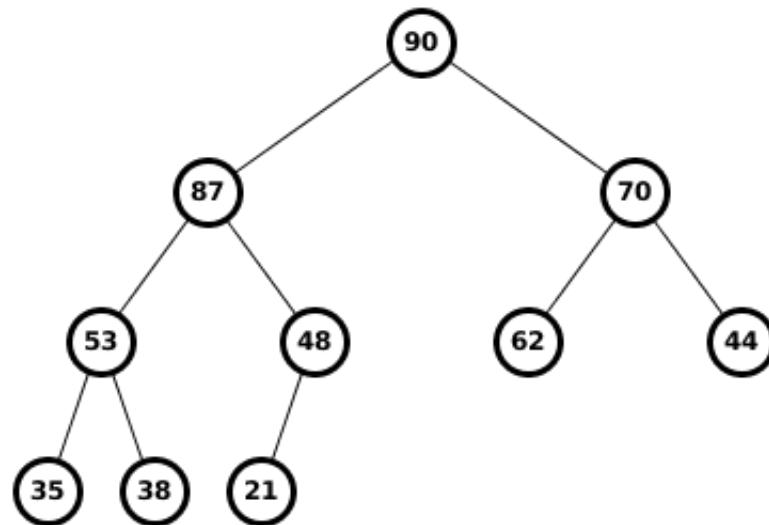
 se $L[i] < menor$ então

$menor := L[i]$

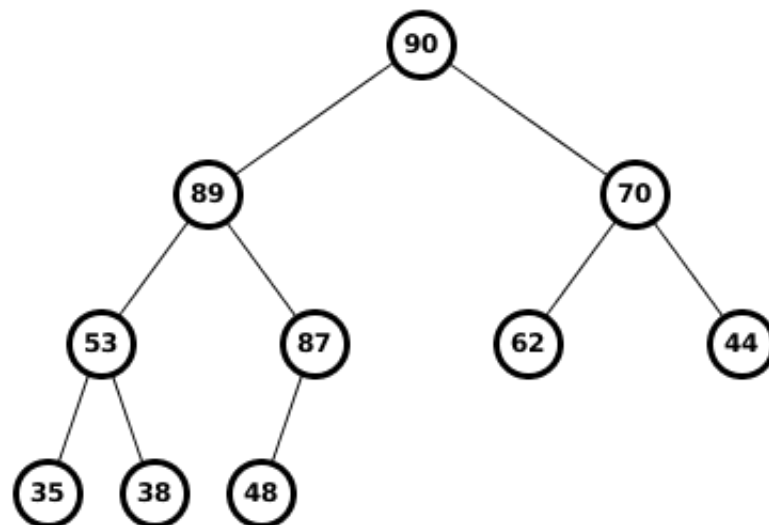
retornar $menor, maior$

3. (2,0) É dada uma fila de prioridade (*heap*) $H = [90 87 70 53 48 62 44 35 38 21]$.
 - (a.) (1,0) Desenhe H em formato de árvore binária após o elemento 21 ter sua prioridade aumentada para 89.

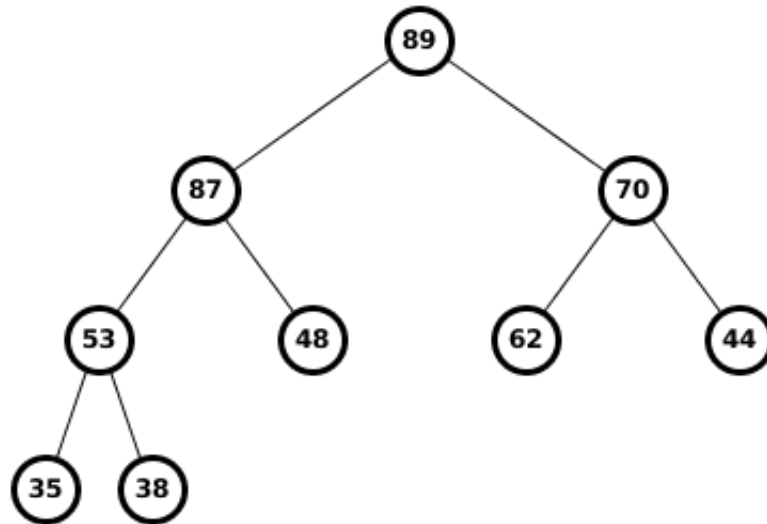
R. Antes da troca de prioridade



Após a troca de prioridade



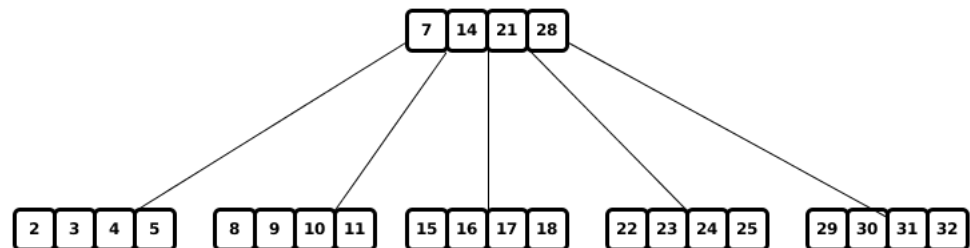
(b.) (1,0) Remova a raiz do *heap* resposta do item (a.), e desenhe o *heap* resultante.



R.

4. (2,0) Responda os seguintes itens:

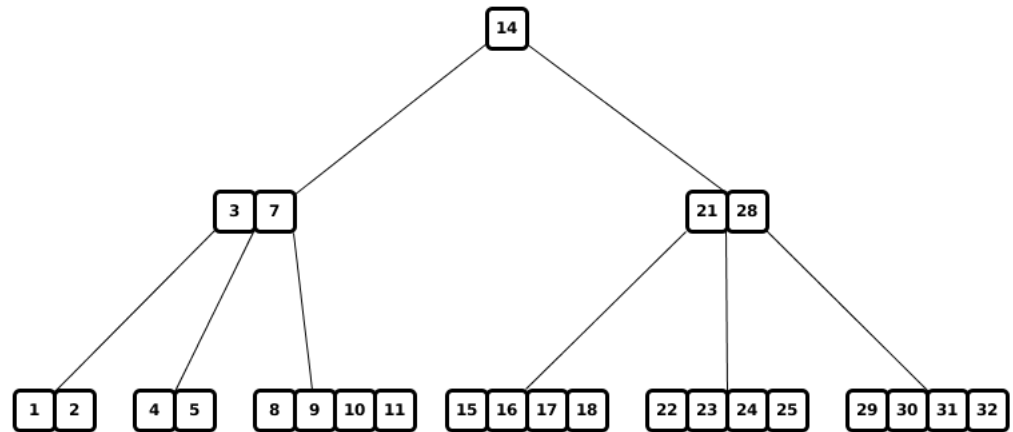
- (a.) (1,0) Desenhe uma árvore B de ordem $d = 2$ e altura $h = 2$ contendo o maior número possível de chaves.



R.

- (b.) (1,0) Insira na árvore B do item (a.) uma chave que seja menor que todas as chaves presentes na árvore, e desenhe a árvore B resultante após a inserção. Diga o nome das operações realizadas durante o processo de inserção.

R. Inserção da chave 1 resulta em duas operações de cisão



5. (2,0) Explique como funciona uma tabela de dispersão onde as colisões são tratadas por encadeamento exterior.

R. Encadeamento exterior é um tipo de tratamento de colisões, que consiste em manter m listas encadeadas, uma para cada possível endereço-base. Um campo para o encadeamento deve ser acrescentado a cada nó. Os nós correspondentes aos endereços-base são apenas nós-cabeça para essas listas. Para buscar uma chave x na tabela T , calcula-se $h(x) = x \bmod m$ e procura-se x na lista encadeada correspondente ao endereço-base $h(x)$. A inclusão de uma nova chave x é feita no final da lista encadeada correspondente ao endereço-base $h(x)$.