

Esta lista pode ser desenvolvida em grupos de até dois alunos, seguindo as especificações contidas no arquivo 00\_ProcListas.pdf disponível na área pública da disciplina na rede. A entrega ao professor deverá ocorrer até o dia **16 de junho de 2021**.

- 1) Adapte o programa que implementa a árvore binária ordenada de inteiros apresentada na apostila, para que ele imprima na tela o conteúdo de cada nó e, após isso, também exiba:
  - Maior valor contido na árvore.
  - Menor valor contido na árvore.
  - Quantidade de folhas da árvore.
  - A altura da árvore.
  - Pesquisa na árvore de um valor informado pelo usuário.

**Obs:** Criar uma função para realizar cada um dos processamentos solicitados acima.

- 2) Segundo as páginas 187 e 188 da Introdução a Estruturas de Dados, de Waldemar Celes e outros, “Para descrever árvores binárias, podemos usar a seguinte notação textual: a árvore vazia é representada por `< >`, e árvores não-vazias, por `<raiz sae sad>`. Com essa notação, a árvore da Figura 13.4 é representada por:

`<a<b< ><d< >> >>><c<e< >> >><f< >> >>>>`

Pela definição, uma subárvore de uma árvore binária é sempre especificada como sendo a *sae* ou a *sad* de uma árvore maior, e qualquer das duas subárvores pode ser vazia. Assim, as duas subárvores da Figura 13.5 são distintas.

Isso também pode ser visto pelas representações textuais das duas árvores que, em **pre-order**, são, respectivamente: `<a<b< >> >>> >> e <a< ><b< >> >>>.`”

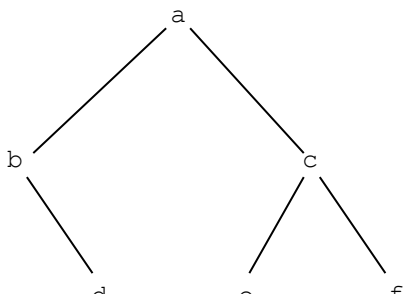


Figura 13.4 Exemplo de árvore binária.

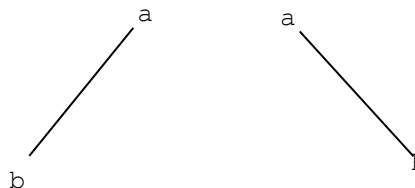


Figura 13.5 Duas árvores binárias distintas

Faça um programa que recebe uma sequência de inteiros e monta a árvore binária ordenada correspondente. Em seguida exibir a descrição da árvore na notação textual indicada anteriormente, porém em sentido **infixo**.

- 3) Faça um programa que implementa uma lista linear duplamente encadeada contendo os dados de cidades informadas pelo usuário. Deverão ser armazenados o nome da cidade (uma palavra de até 30 caracteres úteis), sua população (um inteiro), sua área territorial, PIB e IDH (todos do tipo float). Encerrar a entrada de dados quando o usuário informar “FIM” para o nome do município e imprimir a lista tanto a partir do início como a partir do final. Após a impressão do conteúdo original da lista, o programa deverá executar um *looping* onde, a cada ciclo, um valor de IDH é informado e todas as ocorrências de cidades na lista que possuem IDH igual ou superior a esse valor devem ser excluídas. Após a exclusão, imprimir a lista novamente, tanto a partir do início como a partir do final. Se não for encontrado nenhum nó com valor de IDH igual ou superior ao informado, emitir mensagem apropriada e não imprimir a lista. Encerrar o programa quando o usuário informar um IDH negativo. Organize o programa em arquivo cliente, de interface e de implementação.

## Lista de Exercícios N° 2

- 4) Faça uma versão do programa da lista de cidades criado no exercício 3 desta lista com a implementação sendo por contiguidade. Considere que poderemos ter até 5000 cidades diferentes e, após montada a lista, o programa deverá receber um inteiro  $N$  (entre 1 e 50) correspondente à quantidade de cidades que serão beneficiadas por uma iniciativa do governo direcionada para as localidades de menor IDH. Em seguida, imprimir na tela todos os dados dos  $N$  municípios selecionados. Utilize uma fila de prioridades montada por meio de um MIN-HEAP, com base nos programas de exemplo disponibilizados pelo professor na rede.