

Trabalho 3

Apresentação: 28/11 e 30/11

Conta bancária

O objetivo deste trabalho é estimular os estudantes a se familiarizarem com o comportamento e a lógica associados à estrutura de dados **árvore de busca binária**.

Deenvolvimento de projeto de sistema bancário: lidar com a gestão de registros de clientes (*inserção, remoção e busca*).

Seu formato de entrada e saída estão especificados abaixo, enunciado disponível também no runcodes.

A avaliação consistirá de 3 partes, com suas descrições e pesos a seguir:

1 Implementação e corretude - 40%

Os programas devem ser implementados e submetidos no site <https://runcodes.icmc.usp.br/>, acessando a turma com o código de disciplina GR9C. A nota final dessa parte será dada pela nota obtida nos casos de teste dos exercícios.

Haverão casos de testes ocultos, que serão rodados após o período de submissões, então a nota final pode ser diferente da mostrada na submissão inicial. Realizem também seus próprios testes para garantir a corretude.

2 Organização e boas práticas - 30%

Após o período de submissão os códigos serão revisados buscando boas práticas no desenvolvimento de programas.

Recomendamos o material "Guia de Estilos para ICC I", disponível na página do Tidia.

Além disso,

- Toda a interação com a estrutura deve ser por meio de funções, tais como: alocar, desalocar, imprimir, somar, comparar e outras que o grupo julgar importantes.
- Cada função deve ter um nome descritivo e **fazer somente uma coisa**.
- Documentem o código e deixem comentários pertinentes para seu entendimento, principalmente no começo de funções e loops com lógica importante.
- Desaloquem toda a memória dinamicamente alocada ao fim do código. Dica: usem o `valgrind`.
- Sempre deve-se checar por valores nulos no começo de funções com ponteiros e depois de tentar realizar uma alocação de memória.

3 Seminário - 30%

Deverá ser apresentado um seminário de 15 minutos nos dias 28/11 e 30/11, explicando as implementações realizadas e os resultados obtidos. Todos os integrantes do grupo devem estar presentes.

Por que foi escolhida essa estrutura e não outra?

Como ela resolve o problema?

Como foi implementado?

Quais as suas limitações?

Que dificuldades foram encontradas no caminho da solução?

Referências

Pereira, Leonardo, "*Listas de Exercícios Estruturas de Dados I*", ICMC-USP, 2022.

Conta bancária

Dado um número n de registros bancários, cada um contendo CPF, Nome, Idade e Saldo, estruture esses dados em uma árvore de busca binária e realize a operação indicada.

O primeiro indivíduo deve ser considerado como a raiz da sua árvore, os demais cadastros serão inseridos na árvore por uma regra que no futuro facilitará a busca por pessoas:

Caso o valor numérico do CPF do novo registro de cliente seja maior que o valor do CPF da raiz, ele deve ser inserido na sua sub-árvore direita, caso contrário, na sua sub-árvore esquerda e assim sucessivamente até encontrar a posição adequada.

Lembre que um mesmo CPF não pode estar associado a mais de uma conta dentro do banco.

Após os n registros, seu programa deve receber um 'char' representando uma operação que deverá ser realizada: ['I' : Inserção, 'R' : Remoção, 'B' : Busca]. Receberá então um CPF como parâmetro para todas as operações.

Para a **inserção** de um novo registro, serão passadas as informações da pessoa como foi feito anteriormente na entrada.

Para a **remoção**, utilize quando necessário o antecessor central para decidir que nó será substituído (inorder predecessor).

Entrada

A primeira linha contém um inteiro n , indicando o número de clientes cadastrados no banco. Seguem então n linhas, cada uma contendo um CPF, um nome, uma idade e um saldo, separados por vírgulas. Em seguida uma linha, contendo um caractere, indicando a operação a ser realizada e, por fim, outra linha contendo o CPF de parâmetro (remoção e busca) ou um novo registro (inserção).

Saída

Para os casos de inserção e remoção, deve-se imprimir novamente a árvore na lógica de *preorder*.

Para a busca, seu programa deve apresentar os dados da conta. Caso o CPF em questão não esteja na base de dados, deve-se imprimir a mensagem: *Pessoa nao encontrada*.

Exemplo 1

Entrada	Saída
4 118.469.561-02, Maria Cecilia, 50, 3318.74 498.228.938-76, Julia Lorena, 68, -20278.86 250.627.718-89, Gabriel Maite, 41, 15079.44 904.621.899-61, Livia Isadora, 47, 9113.10 B 904.621.899-61	904.621.899-61, Livia Isadora, 47, 9113.10

Exemplo 2

Entrada	Saída
4 118.469.561-02, Maria Cecilia, 50, 3318.74 498.228.938-76, Julia Lorena, 68, -20278.86 250.627.718-89, Gabriel Maite, 41, 15079.44 025.801.967-06, Samuel Marques, 23, -3041.86 I 042.345.357-22, Maria Rita, 19, 101120.23	118.469.561-02, Maria Cecilia, 50, 3318.74 025.801.967-06, Samuel Marques, 23, -3041.86 042.345.357-22, Maria Rita, 19, 101120.23 498.228.938-76, Julia Lorena, 68, -20278.86 250.627.718-89, Gabriel Maite, 41, 15079.44

Exemplo 3

Entrada	Saída
10 118.469.561-02, Maria Cecilia, 50, 3318.74 498.228.938-76, Julia Lorena, 68, -20278.86 250.627.718-89, Gabriel Maite, 41, 15079.44 904.621.899-61, Livia Isadora, 47, 9113.10 025.801.967-06, Samuel Marques, 23, -3041.86 539.696.899-02, Joaquim Samuel, 0, 6868.53 782.819.678-46, Gabriel Jesus, 47, 12237.87 370.986.568-95, Maria Julia, 60, -14561.91 550.067.540-01, Arthur Lorena, 19, -8101.50 992.090.680-84, Isaac Fernandes, 50, 11898.13 R 904.621.899-61	118.469.561-02, Maria Cecilia, 50, 3318.74 025.801.967-06, Samuel Marques, 23, -3041.86 498.228.938-76, Julia Lorena, 68, -20278.86 250.627.718-89, Gabriel Maite, 41, 15079.44 370.986.568-95, Maria Julia, 60, -14561.91 782.819.678-46, Gabriel Jesus, 47, 12237.87 539.696.899-02, Joaquim Samuel, 0, 6868.53 550.067.540-01, Arthur Lorena, 19, -8101.50 992.090.680-84, Isaac Fernandes, 50, 11898.13