

USP – ICMC – SSC

SSC0603 – Estrutura de Dados I (ED1) – PROVA PF

Professor responsável: *Fernando Santos Osório*

Semestre: 2023/2

Horário: Terça 14h20 e Quinta 14h20

Wiki: SSC-603-2023 e **Facebook:** SSC0603-FOsorio

Web: <http://www.icmc.usp.br/~fosorio/>

NRO. USP: < Colocar o seu NUSP comentado no **programa fonte**>

DATA: 12 / 12 / 2023

NOME : < Colocar o seu Nome comentado no **programa fonte**>

>> COLOCAR SEU **NOME** E **NRO. USP** COMO COMENTÁRIO DO PROGRAMA ENTREGUE

>> O PROGRAMA DEVE SER ENVIADO PARA O RUNCODES.ICMC com NRO USP NO CÓDIGO FONTE

PROVA FINAL – SSC0603 ED1

Implemente o seguinte programa descrito abaixo de acordo com o especificado USANDO UMA ÁRVORE BINÁRIA ORDENADA e BALANCEADA (p.ex. Árvore AVL) para armazenar os dados manipulados no programa. Se você não conseguir usar uma ABB+Balanceada, pode usar uma ABO “normal”, porém será descontada uma parte da nota (-1.0 pto na nota máxima).

Portanto, a prova poderá ser feita usando o TAD de AVL (se não der, use o TAD ABO)

Poderá ser usado o TAD do Livro do Backes ou qualquer outro TAD de sua preferência.

O programa deve ser bem “estruturado e organizado” (Uso de TAD, modular).

O programa TEM que usar Estruturas de Dados do tipo Árvore Binária Ordenada (Balanceada)

Valores INTEIROS armazenados: coordenadas de pontos X,Y,Z (valores inteiros) armazenados em 3 árvores, uma árvore ordenada por X, uma outra por Y, e por fim uma outra por Z.

O programa deve ler de um arquivo de entrada denominado de “dados.txt” os dados que descrevem uma “nuvem de pontos X Y Z” (cenário, objeto 3D). Os dados são coordenadas, representadas por três valores inteiros (X,Y,Z). Definições:

- Os nodos possuem 3 dados (X,Y,Z inteiros) em todas 3 árvores (pode ser usado um destes 3 valores, duplicado, para usar como valor de ordenação), neste caso o nodo teria Valor + X + Y + Z;
- Estes dados deverão ser armazenados de forma ordenada em 3 árvores, sendo cada árvore considerando como “dado principal” (usado para a ordenação), em cada árvore, cada vez usando um dos três valores para ordenar.
- Por exemplo, uma primeira árvore contém (X,Y,Z) mas é ordenada por X (dado principal). A segunda árvore contém os dados (X,Y,Z) mas é ordenada por Y (dado principal). A terceira árvore contém os dados (X,Y,Z) mas é ordenada por Z.

- Isso vai permitir que se localize RAPIDAMENTE o X mais próximo (menor valor da árvore X), o Y mais próximo (menor valor da árvore Y) e o Z mais próximo (menor valor da árvore Z) [em 3D].

ARQUIVO:

- TODOS os valores lidos devem ser lidos de um ARQUIVO TEXTO com nome “dados.txt” (o arquivo se encontra no diretório corrente, SEM PATH específico). Cada linha do arquivo contém 3 números inteiros separados por espaço, e cada nova linha contém mais 3 novos valores inteiros do próximo ponto 3D. O arquivo termina sempre pelo ponto “0 0 0” (coordenadas 0,0,0), terminando a lista de dados (pontos 3D) ao ser lida esta última linha do arquivo.

- TODOS os valores lidos devem ser inseridos nas árvores. Cada ponto (X,Y,Z) é inserido em cada uma das três árvores.

Exemplo de Arquivo de Dados de Entrada:

```
1 10 100
2 20 200
3 30 300
4 40 400
5 50 500
0 0 0
```

Na primeira linha a coordenada 3D é X=1, Y=10 e Z=100, sendo o ponto XYZ (1,10,100).

O arquivo, como pode ser visto acima, termina na coordenada “0 0 0” (último ponto).

Este ponto (0,0,0) NÃO DEVERÁ ser inserido nas árvores!

UMA VEZ CRIADAS AS ÁRVORES:

Ler um valor inteiro do TECLADO, e conforme este dado lido executar as seguintes operações abaixo, sendo que o dado é um valor 1, 2, 3 ou 4

1 - Exibe a altura da árvore ordenada por X e a quantidade de nodos. Exemplo:

Entrada pelo Teclado: 1

Saída na Tela: 3 5 // No Exemplo acima a árvore tem altura 3 e 5 nodos.

2 – Lê mais um valor inteiro, buscando esse valor na árvore ordenada por Z.

Exibe o valor em Z de cada nodo visitado até achar ou não o valor (mostra caminho).

Se achou, exibe “YES” se não achou exibe “NO”

Entrada pelo Teclado: 2 300

Saída na Tela: 200 400 YES // Note que se a AVL ficou diferente a saída pode ser outra

3 – Lê um dado (valor inteiro) e busca o sucessor desse dado na árvore ordenada em Y.

Entrada pelo Teclado: 3 20

Saída na Tela: 30

4– Busca o menor dado na árvore ordenada em X e exibe.

Busca o menor dado na árvore ordenada em Y e exibe.

Busca o menor dado na árvore ordenada em Z e exibe.

Entrada pelo Teclado: 4

Saída na Tela: 1 10 100

That's all folks!

ENTREGA DA PROVA:

- A prova deve ser enviada no **RunCodes.ICMC** (site: <https://runcodes.icmc.usp.br/> no seu nome)
Na turma SSC0603 Estrutura de Dados I - 2023-2 (nossa turma que estamos usando atualmente)
- O Exercício do RunCodes é o “PROVA FINAL – ED1 2023 – Fosorio
- Enviar um programa fonte único (programa .c) OU um arquivo .ZIP com Makefile e os fontes no padrão aceito pelo RunCodes.
- Não esqueça de colocar o seu nome como comentário nos programas fontes
- Se der problema com o envio para o RunCodes (**apenas se não conseguir enviar**), avise o professor e envie por e-mail para: fosorio@icmc.usp.br e com cópia para fosorio@gmail.com

BOA PROVA!

REGRAS EM RELAÇÃO REALIZAÇÃO DESTA PROVA

1. A PROVA É **INDIVIDUAL** com consulta Papel (Livros, papel impresso ou escrito) e formato Digital (Internet, Wiki, Pendrive), porém **SEM CONSULTAR** ou **SE COMUNICAR COM HUMANOS** ou **QUALQUER OUTRA FORMA DE VIDA TERRESTRE** ou **EXTRA-TERRESTRE!**
Não podem usar de formas de comunicação com pessoas externas, além do professor, referente a prova, seja por celular (manter desligado/guardado), por e-mail, por whatsapp, por mensagens ou fóruns, com colegas, etc.
NÃO É PERMITIDO O EMPRÉSTIMO DE MATERIAL (Cadernos, Anotações, Livros, etc).
2. A PROVA DEVE OBRIGATORIAMENTE SER FEITA NO COMPUTADOR DO LABORATÓRIO.
Não é permitido uso de celular, notebook ou tablet na prova, USE APENAS O DESKTOP do LAB.
3. A PROVA DEVE SER ENTREGUE ENVIANDO NO RunCodes.ICMC
Ver detalhes sobre a “ENTREGA DA PROVA” acima.
4. A PROVA USUALMENTE DEVE SER REALIZADA USANDO O CODEBLOCKS DO LABORATÓRIO, se for usar outro Compilador/IDE, INDIQUE NO E-MAIL qual AMBIENTE, COMPILADOR e IDE usou!
5. AO ENTREGAR A PROVA O(A) ALUNO(A) CONCORDA COM ESTAS REGRAS E SE COMPROMETE A FAZER A PROVA INDIVIDUALMENTE!

FIM
