

| | | |
|--|--------------------------------|---|
| Disciplina DCE792 - AEDS 2 | Método de realização Código | Data de apresentação 23/10/2024 às 23h59 |
| Professor Iago Augusto de Carvalho (iago.carvalho@unifal-mg.edu.br) | | |

Trabalho prático 2 - Comparação entre árvores balanceadas e não-balanceadas

Como estudado na disciplina, árvores podem sofrer um grande desbalanceamento. Desta forma, o objetivo deste trabalho é verificar o quão lento é uma árvore não balanceada em comparação a uma árvore balanceada.

Neste trabalho, cada grupo de dois ou três estudantes deverá árvores balanceadas e não-balanceadas e comparar seu tempo de processamento.

O que deve ser desenvolvido: Cada grupo (de dois ou três integrantes) deverá implementar dois algoritmos. O primeiro será uma árvore binária não-balanceada, e o segundo uma árvore balanceada qualquer a escolha do grupo. O código deverá ser desenvolvido em linguagem de programação *C*, não sendo permitido o uso de bibliotecas externas ou bibliotecas *STL* do *C++*.

O código deverá contabilizar o tempo, em milissegundos, gastos para realizar todas as operações de inserção e remoção descritas nas instâncias dadas.

Entradas: Diversas instâncias de teste serão dadas como entrada. Cada instância de teste será composta por duas colunas. A primeira coluna será um caractere **I** ou um caractere **R**, enquanto a segunda coluna será um número inteiro. O caractere **I** significa uma inserção na árvore, enquanto o caractere **R** simboliza uma remoção. O inteiro a ser inserido ou removido será aquele que consta na coluna logo a frente.

Saída esperada: Espera-se que o código imprima somente dois números em ponto flutuante. O primeiro número deverá indicar o tempo necessário para realizar as operações de inserção e remoção na árvore não balanceada, enquanto o segundo deverá simbolizar o tempo das mesmas operações de inserção e remoção na árvore balanceada. A saída deverá, necessariamente, ser formatada em duas linhas, sendo que

- A primeira linha indica o tempo da árvore binaria não-balanceada
- A segunda linha indica o tempo da árvore balanceada

Código base: O código base deste trabalho está disponível em um repositório do GitHub Classroom¹ onde todos os alunos já foram associados. Este diretório contém o código-base para a realização do trabalho prático assim como as instâncias de teste. O código-base contém exemplos de como medir o tempo de execução, inclui um *makefile*, além de exemplificar a maneira como a saída deverá ser exibida. Cada grupo é livre para modificar o código-base da maneira que entender.

Makefile: O trabalho deverá, **obrigatoriamente**, compilar com um arquivo *makefile*, sendo que um arquivo de exemplo é disponibilizado junto do código-base deste trabalho. Este *makefile* pode ser modificado caso o grupo julgue necessário.

O código deve compilar em um ambiente Linux padrão, como o disponível nos laboratórios da UNIFAL. O código deve compilar com o comando

make all

Observa-se que o trabalho que não compilar com o *makefile* disponibilizado levará nota **zero**.

¹<https://classroom.github.com/a/K7.2VYVB>

Entrega e avaliação: Cada grupo deverá clonar o repositório dado. A entrega será realizada através do próprio GitHub Classroom, onde será necessário realizar o *commit* e o *push* dos algoritmos desenvolvidos.

Deverá ser entregue somente um repositório por grupo. O repositório deverá conter *.c*, todos os arquivos *.h*, e o *makefile* utilizado pelo grupo. O *push* deverá ser realizado até as 23h59 do dia 23/10/2024. Entregas em atraso não serão toleradas.

A nota do trabalho será um valor z entre 0 e 100, sendo

$$z = 100 - c - d$$

onde c é um binário que indica se o código compila ou não utilizando o *makefile* enviado e d é um valor real entre 0 e 1 que representa a qualidade do código de acordo com os parâmetros abaixo definidos.

Método de avaliação: Somente o código será avaliado. Os parâmetros a serem avaliados são

- A qualidade e clareza do código
- Comentários explicativos
- Execução correta dos algoritmos
- Saída correta de acordo com a proposta
- Facilidade de uso do *makefile*