

← SSC0300: Trabalho 02- Aplicações de Algoritmos de Árvore

SSC0300 - Linguagens de Programação e Aplicações

Prof. Adenilso da Silva Simão

Trabalho 2

Entrega: 05/11/2015

Aplicações de Algoritmos de Árvores

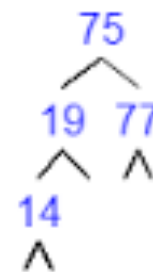
Utilizando o conhecimento adquirido durante as aulas sobre algoritmos de árvore resolva os seguintes problemas usando a linguagem de programação C:

1. **Operações Básicas em Árvores (5 pts)** - Implementar um programa que dada uma sequência de números inteiros armazene-os em uma Árvore Binária de Busca (ABB) e permita a realização de cinco operações:

- **Inserção:** O programa deve receber via entrada padrão um valor inteiro que defina o total de números inteiros (positivos e/ou negativos) a serem lidos em sequência posteriormente também via entrada padrão.
- **Remoção:** O programa deve receber via entrada padrão a chave de um valor a ser removido.
- **Busca:** O programa deve permitir a consulta de valores. Ao ser buscado um valor, o programa deve reportar a existência ou ausência do mesmo na ABB.
- **Impressão da árvore em:** Pré Ordem, Pós Ordem, e Em Ordem.
- **Impressão da árvore em “labelled bracketing”:** A notação *labelled bracketing* é um modo alternativo para representar árvores usando colchetes. Cada nó é representado usando pares de colchetes. A primeira string dentro de um par de colchets define o rótulo de um nó e os elementos consecutivos os nós filhos. E.g.: Usando a notação *labelled bracketing*, uma árvore binária gerada a partir dos elementos **75, 19, 77 e 14** pode ser representada como mostrado a seguir:

[75 [19 [14 [] []] []] [77 [] []]]

Representação em “labelled bracketing”



Representação Gráfica

Essa notação pode ser usada como entrada nas ferramentas [syntree](#)^[1], [phpsyntaxtree](#)^[2] e [rsyntaxtree](#)^[3]. Lembrando que por se tratar de uma árvore binária cada nó deverá ter no máximo dois filhos. O exemplo acima pode ser visualizado no link no rodapé da página^[4].

← SSC0300: Trabalho 02- Aplicações de Algoritmos de Árvore

simule uma árvore genealógica. Assuma que cada casal tem no máximo um filho. O programa deve receber um conjunto de N tuplas de 3 nomes representando relações entre filho, pai e mãe, e.g. “**filho1 pai1 mãe1**” onde **pai1** e **mãe1** são nós filhos de **filho1**. Além disso, as seguintes operações devem ser disponibilizadas:

- Imprimir membros de uma família por geração
 - Imprimir os antepassados de um dado indivíduo
 - Imprimir árvore genealógica usando “labelled bracketing”
 - Calcular o número equivalente ao grau de parentesco entre membros da família.
- i. E.g.: Dado “**filho1 pai1 mae1**”, “**pai1 avô1 avó1**”, “**mae1 avô2 avo2**” e “**avo1 bisavô1 bisavó1**” temos que

- entre **filho1** e **pai1** grau de parentesco 1
- entre **mae1** e **pai1** grau de parentesco 1
- entre **filho1** e **avo1** grau de parentesco 2
- entre **filho1** e **bisavo1** grau de parentesco 3

Caso não exista uma relação direta de parentesco, tal como entre **avô1** e **avó2** deve se reportar “entre **avô1** e **avó2** grau de parentesco 0”

3. **Conversor de Equação Polonesa (Ponto Extra: +3 pts)** - Implementar um programa que dada uma equação em notação polonesa apresente suas equivalentes usando a notação polonesa reversa (RPN) e infixa. **CUIDADO COM OS PARÊNTESES DA NOTAÇÃO INFIXA!** Abaixo Segue um exemplo:

Entrada:

* - 5 6 7

Saída:

RPN: 5 6 - 7 *
Notação Infixa: ((5 - 6) * 7)

Ao serem executados, os programas referentes aos problemas 1 e 2 devem pela entrada/saída padrão apresentar um menu com as opções descritas acima, devem permitir as operações de inserção, busca e remoção de elementos. O mesmo vale para o problema 3 que deve permitir a conversão de equações para a notação infixa e RPN. **Cada um dos programas deve ser implementado separadamente e contar com uma entrada-exemplo!**

Entrega

Deve ser produzido um **relatório** em formato pdf contendo as seguintes seções:

1. **Introdução** descrevendo o trabalho, membros da equipe
2. **Descrição do projeto** apresentando os artefatos relacionados ao trabalho (e.g. ambiente de desenvolvimento, compilador usado, parâmetros de compilação, e plataforma de compilação da versão binária (e.g. x86, x64), códigos fonte (.c), bibliotecas (.h), versões compiladas, entradas-exemplo, imagens das árvores...)
3. **Tutorial** com instruções de como compilar e executar cada um dos programas
 - a. Obs.: **cenários de uso** com **entradas-exemplo** definidas pelo(s) aluno(s) devem ser apresentados para cada uma das operações.
4. **Outras Informações**, tais como bugs e limitações do programa, podem ser inseridas em uma seção opcional no final do relatório.

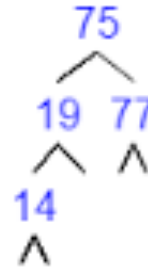
Todos os artefatos, tais como códigos fonte (.c), bibliotecas (.h), entradas-exemplo e o relatório, devem ser entregues junto com uma versão compilada. Além disso, deve-se incluir um arquivo de **README** com uma visão geral do conteúdo do repositório e responsáveis pelo trabalho. Para trabalhos feitos em grupo, incluir nome e matrícula dos membros na ordem de pontuação.

← SSC0300: Trabalho 02- Aplicações de Algoritmos de Árvore

isso, o aluno devera fazer uma postagem no facebook contendo o link para o projeto no github, o nome da tag a ser avaliada e o nome dos alunos participantes.

Exemplos de entrada para os problemas:

• Problema 1



- Representação Gráfica da árvore ao lado
- Entrada (4 elementos):

4
75
19
77
14

- Impressão Em ordem

14 19 75 77

- Impressão Pré ordem

75 19 14 77

- Impressão Pós ordem

14 19 77 75

- Impressão em "labelled bracketing"

[75 [19 [14 [] []] []] [77 [] []]]

• Problema 2

- Entrada (4 entradas)

4
filho1 pai1 mae1
pai1 avô1 avó1
mae1 avô2 avo2
avo1 bisavô1 bisavó1

- Imprimir membros de uma família por geração

■ Saída:

filho1
pai1 mae1
avô1 avó1 avô2 avó2
bisavô1 bisavó1

- Imprimir os antepassados de um dado indivíduo

■ Entrada: **pai1**

■ Saída: **avô1 avó1 bisavô1 bisavó1**

- Imprimir árvore genealógica usando "labelled bracketing"

■ **[filho1 [pai1 [avô1 [bisavô1] [bisavó1]] [avó1]] [mae1 [avô2] [avo2]]]**

■ **Ilustração:**

← SSC0300: Trabalho 02- Aplicações de Algoritmos de Árvore



- Calcular o número equivalente ao grau de parentesco entre membros da família.
 - Entrada: ***pai1 mae1***
 - Saída: ***entre pai1 e mae1 grau de parentesco 1***
 - Entrada: ***pai1 filho1***
 - Saída: ***entre pai1 e filho1 grau de parentesco 1***
 - Entrada: ***avô1 avó2***
 - Saída: ***entre avô1 e avó2 grau de parentesco 0***

[1] <http://mshang.ca/syntree/>

[2] <http://www.ironcreek.net/phpsyntaxtree/>

[3] <http://yohasebe.com/rsyntaxtree/>

[4] <http://goo.gl/4dTWxV>