

Trabalho 02: Aplicações de Algoritmos em Árvores

SSC0300 – Linguagens de Programação e Aplicações

Alunos:

- 1- Felipe Augusto Senger Lopes de Souza - N° USP: 5513317
- 2- Lucas Souza de Oliveira - N° USP: 9312515
- 3- Gustavo de Paula Lourenço - N° USP: 9312842

Introdução: Este projeto contém os 3 exercícios do 2º Trabalho da disciplina de Linguagens de Programação e Aplicações. Eles são programas em C que utilizam os conceitos de Algoritmos em Árvores para resolver os problemas propostos.

Descrição do Projeto: Todos os programas foram desenvolvidos no Windows 7 64 bits, no programa Dev C++. O compilador utilizado foi o default do Dev C++, v5.10, nas configurações TDM-GCC 4.8.1 64-bit Release. Todas as bibliotecas usadas foram as consideradas padrão do Dev C++. A compilação não requer nenhum parâmetro especial e os programas foram compilados usando o botão de compilação do Dev C++.

Segue a descrição de como os programas funcionam e são executados:

Exercício 1 – Operações Básicas em Árvores:

O programa recebe a entrada de uma quantidade n de números inteiros e os armazena em uma Árvore Binária de Busca (ABB). Além disso, ele permite a inserção de novos números, a remoção de números (inclusive da raiz), a busca de números, sua impressão Em Ordem, Pré Ordem e Pós Ordem e sua impressão em Labelled Bracketing.

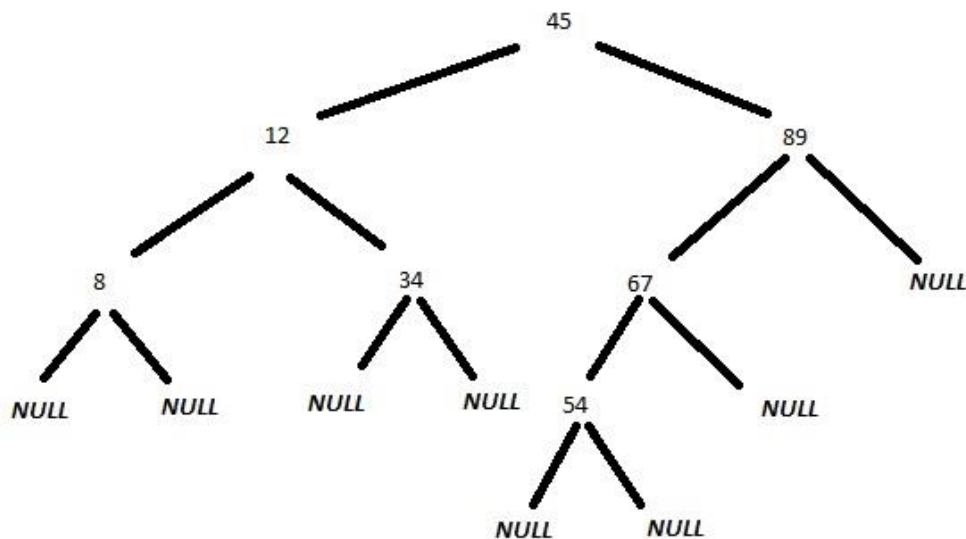
Para que tudo isso seja feito, foram criadas diferentes funções e o programa conta com um menu, que pergunta ao usuário o que ele deseja fazer, e chama as determinadas funções. Para inserir números, o programa chama a função de inserção para cada número, criando os nós um por um (alocando a memória um por um com o uso do *malloc*). Para remover números, existem duas funções, devido ao caso específico da remoção da raiz, ou seja, existe uma função para remover a raiz e outra para remover outros números.

Quanto à organização da Árvore, o programa organiza o número a partir de uma raiz. Números menores são inseridos à esquerda de números maiores, e números maiores à direita, e assim a árvore fica ordenada (números iguais são colocados à esquerda, para que o programa consiga trabalhar também com números iguais). É importante ressaltar que por se tratar de uma árvore binária, cada nó possui no máximo dois ramos. Para montar a árvore é preciso dizer ao programa quantos números serão inseridos, e depois os números a serem organizados. Notar que o primeiro número será a raiz, e os demais serão organizados da forma descrita acima.

Exemplo: Caso o usuário entre com a seguinte sequência

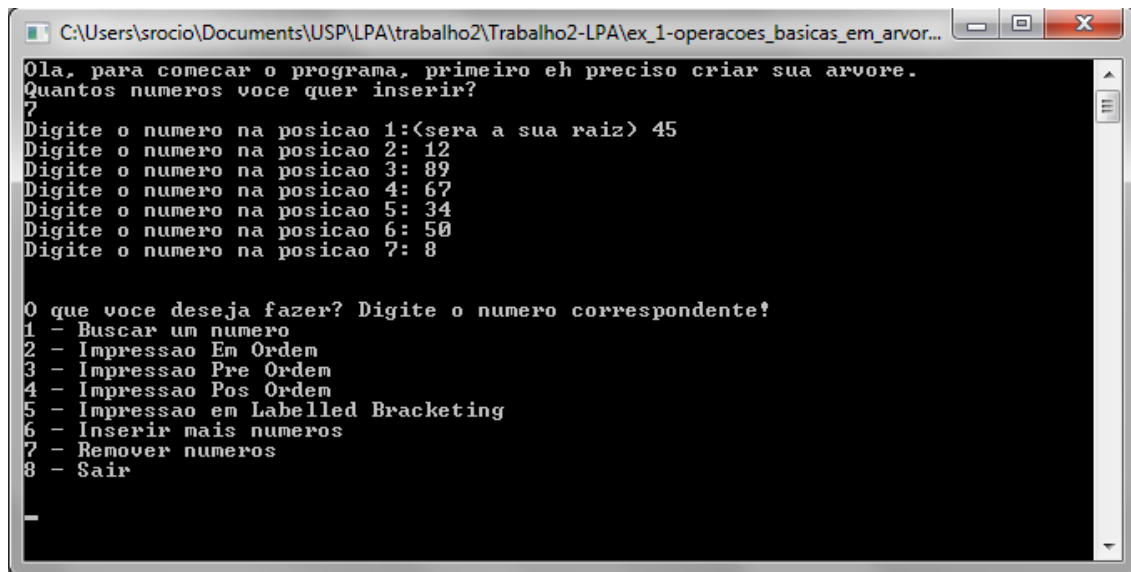
7
45
12
89
67
34
50
8

Uma árvore como a ilustrada a seguir seria organizada pelo programa:



Árvore-exemplo - exercício 1

Tutorial de uso: O programa deve ser compilado nas configurações citadas acima, que são as consideradas “default” do Dev C++ (versão 5.10). O primeiro número a ser entrado é a quantidade n de números que o usuário quer inserir na árvore, em seguida, deve-se inserir os n números, sabendo que o primeiro será a raiz da árvore (caso seja fornecido um número menor do que 1, o programa irá pedir que o usuário forneça um novo número). Feita a inserção, um menu irá aparecer, para que o usuário possa escolher o que deseja que o programa faça:



```
C:\Users\srocio\Documents\USP\LPA\trabalho2\Trabalho2-LPA\ex_1-operacoes_basicas_em_arvor...
Ola, para começar o programa, primeiro eh preciso criar sua arvore.
Quantos numeros voce quer inserir?
7
Digite o numero na posicao 1:(sera a sua raiz) 45
Digite o numero na posicao 2: 12
Digite o numero na posicao 3: 89
Digite o numero na posicao 4: 67
Digite o numero na posicao 5: 34
Digite o numero na posicao 6: 50
Digite o numero na posicao 7: 8

0 que voce deseja fazer? Digite o numero correspondente!
1 - Buscar um numero
2 - Impressao Em Ordem
3 - Impressao Pre Ordem
4 - Impressao Pos Ordem
5 - Impressao em Labelled Bracketing
6 - Inserir mais numeros
7 - Remover numeros
8 - Sair
```

Menu - exercício 1

Basta digitar o número e apertar ‘Enter’, por exemplo, se deseja obter a impressão da árvore em “Labelled Bracketing”, deve-se digitar **5** e depois teclar ‘Enter’.

Exercício 2 – Árvore genealógica:

O programa recebe a entrada de uma quantidade n de tuplas, ou seja, um grupo de nomes da família, na ordem “filho mãe pai”, e os armazena numa árvore binária (fazendo a consideração de que cada casal possui apenas 1 filho). Além disso, o programa permite a impressão de cada geração, impressão dos antepassados de um dado indivíduo, impressão da árvore em “labelled bracketing” e o cálculo do grau equivalente de parentesco entre duas dadas pessoas, além de permitir a inserção de mais membros na árvore.

Para que tudo isso seja feito, foram criadas diferentes funções e o programa conta com um menu, que pergunta ao usuário o que ele deseja fazer, e chama as determinadas funções. Para inserir as “tuplas”, o programa chama a função de inserção para cada tupla, criando os nós um por um (alocando a memória um por um com o uso do *malloc* para cada nome dado). Para fazer o controle do “nível” de cada indivíduo na árvore, cada pessoa possui um grau. O filho tem grau 0, os pais grau 1, avós grau 2, e assim por diante. Esse grau é utilizado para calcular o grau de parentesco entre duas pessoas (por meio de uma subtração entre os dois graus) e para imprimir as gerações da árvore. A função que imprime as gerações da árvore faz uso de um algoritmo de percorrimto da árvore Em Ordem, pois é necessário procurar na árvore toda as pessoas de cada grau, podendo imprimir as determinadas gerações. Também foi feita uma função recursiva para a impressão em “labelled bracketing”, e uma função *select*, muito utilizada na execução do programa, pois ela é capaz de localizar um dado nome na árvore, e retornar o seu endereço, para que ele possa ser utilizado em diferentes funções como a de inserção de números e a do cálculo do grau de parentesco equivalente.

Quanto à organização da árvore, o primeiro filho inserido é a raiz. Os dois ramos que saem a partir dele são seus pais, os dois ramos de cada pai os avós, e assim por diante. É

preciso fornecer ao programa quantas “tuplas” serão entradas, cada uma contendo um filho e seus pais. É importante notar que a partir da segunda “tupla”, o nome do filho deve já pertencer à árvore, e, portanto o usuário deve fornecer o nome do filho de quem se quer dar o nome dos pais, e depois o nome dos pais.

Exemplo: Para uma entrada como a seguinte

4

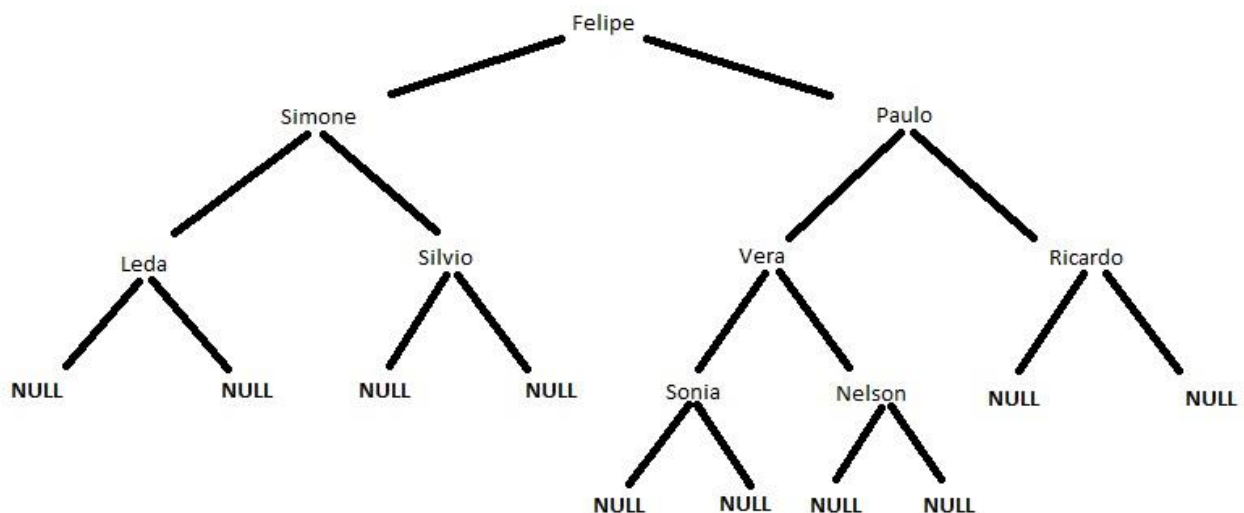
Felipe Simone Paulo

Paulo Vera Ricardo

Simone Leda Silvio

Vera Sonia Nelson

O programa irá organizar a árvore ilustrada a seguir:



Árvore-exemplo - exercício 2

Tutorial de uso: O programa deve ser compilado nas configurações citadas acima, que são as consideradas “default” do Dev C++ (versão 5.10). O primeiro número a ser entrado é a quantidade n de “tuplas” que o usuário quer inserir na árvore (caso seja fornecido um número menor do que 1, o programa irá pedir que o usuário forneça um novo número), em seguida, deve-se inserir os n trios de nomes, sabendo que o primeiro filho será a raiz da árvore. Em seguida, na próxima entrada, deve-se fornecer o nome de alguém que se quer inserir os pais, e depois os pais deste indivíduo. Portanto, a partir da segunda tupla, os filhos entrados já devem existir na árvore, e com essa tupla serão inseridos seus pais. Feita a inserção, um menu irá aparecer, para que o usuário possa escolher o que deseja que o programa faça:

```
C:\Users\srocio\Documents\USP\LPA\trabalho2\Trabalho2-LPA\ex_2-arvore-genealogica.exe
Ola, este programa ira montar uma arvore genealogica.
Para inserir as pessoas, digite seus nomes, de 3 em 3, separados por espaco, na
seguinte ordem: 'filho mae pai' e depois teclar 'Enter'
Quantas dessas tuplas (grupos de 3 nomes) voce quer entrar?4
Digite a tupla 1 (a arvore toda sera feita a partir deste filho):
Felipe Simone Paulo
Digite a tupla 2:
Paulo Vera Ricardo
Digite a tupla 3:
Simone Leda Silvio
Digite a tupla 4:
Vera Sonia Nelson

0 que voce deseja fazer? Digite o numero correspondente!
1 - Imprimir membros da arvore por geracao
2 - Imprimir os antepassados de um individuo
3 - Impressao da arvore em 'labelled bracketing'
4 - Calcular o grau de parentesco entre 2 membros
5 - Inserir mais individuos
6 - Sair
```

Menu – exercício 2

Basta digitar o número e apertar 'Enter', por exemplo, se deseja obter a impressão da árvore por gerações, deve-se digitar **1** e depois teclar 'Enter'.

Observações importantes:

- É importante seguir as instruções dadas pelo programa no início, as tuplas devem estar na ordem “filho mae pai”, caso contrário, a árvore será organizada incorretamente.
- Cada nome pode conter até **40 caracteres**.
- Cada casal pode possuir apenas 1 filho.
- O programa não funciona com nomes iguais.
- Após a entrada de cada tupla deve ser teclado 'Enter'.
- Os nomes devem ser separados por um espaço simples.
- Os nomes entrados devem ser simples, e caso seja necessário o uso de um nome composto, ele deve ser digitado sem espaços para que não ocorram erros na inserção de nomes na árvore.

Exercício 3 –Conversor de Equação Polonesa

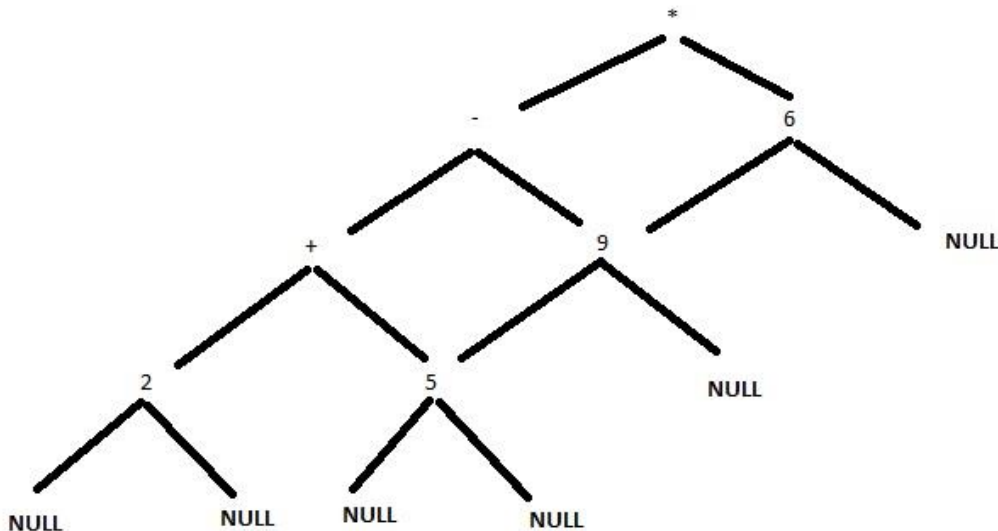
O programa recebe a entrada de uma equação em notação polonesa e a transforma para a notação polonesa reversa (RPN) e para a notação infixa. Para isso, o programa organiza a equação polonesa dada em uma árvore, e seguindo algoritmos de impressão, consegue convertê-la para a RPN (por meio da impressão Pós Ordem) ou para a notação infixa (por meio da impressão em ordem).

Para isso, o programa conta com duas funções de inserção e duas funções de percorrimento: as duas funções de inserção realizam a inserção de números e sinais separadamente. A função *insertsign* insere todos os sinais à esquerda da árvore, e a função *insertnum* insere os números nos locais corretos dependendo da operação que realizam (um

exemplo será mostrado a seguir). As duas funções de percorrimento, *posordem* e *emordem*, servem para a conversão da equação para as notações RPN e infixa, respectivamente.

Exemplo: Para a entrada da equação em notação polonesa a seguir, o programa organizará a seguinte árvore:

Entrada: * - + 2 5 9 6



Árvore-exemplo – exercício 3

Tutorial de uso: Ao inicializar o programa, ele pedirá que o usuário entre com sua equação na notação polonesa. É importante seguir as instruções de preenchimento para que o programa funcione, separando cada número por um espaço e teclando 'Enter' no final (caso deseje, o usuário poderá teclar 'Enter' entre cada entrada como uma opção alternativa de preenchimento). Feita a inserção, um menu irá aparecer, para que o usuário possa escolher o que deseja que o programa faça:

```
C:\Users\srocio\Documents\USP\LPA\trabalho2\Trabalho2-LPA\ex_3-conversor-de-equacao-polon...
Ola! Este programa ira converter uma entrada em notacao polonesa para a notacao
polonesa reversa (RPN) e infixa.
Digite sua equacao em notacao polonesa
(separe os simbolos e numeros todos por um 'espaco' e depois tecle 'Enter'):
```

```
* - + 2 5 9 6

0 que voce deseja fazer? Digite o numero correspondente!
1 - Impressao na notacao polonesa reversa (RPN)
2 - Impressao na notacao infixa
3 - Sair
```

Basta digitar o número e apertar 'Enter', por exemplo, se deseja obter a conversão para notação polonesa reversa, deve-se digitar **1** e depois teclar 'Enter'.

Observações importantes:

- O programa pode receber apenas as operações usuais (*, /, +, -).
- O programa aceita apenas os algarismos de 0 a 9.
- O cálculo do valor final da expressão não é realizado, o programa é capaz apenas de fazer a conversão entre as notações.