

# Laboratório 08

## Diâmetro de Árvore Binária

### MC202 - Estruturas de Dados

25 de outubro de 2018

## 1 Especificação

A função básica das redes de comunicação de hoje é a transmissão de pacotes de dados entre computadores, processadores ou outros periféricos. A Figura 1 ilustra uma árvore binária completa com 3 entradas e 3 saídas. Os quadrados representam nós terminais, origem ou destino dos pacotes de dados, enquanto os círculos representam os *switches* de comunicação que direcionam os pacotes na rede. Um *switch* recebe pacotes por meio das ligações de entrada e encaminha-os pelas ligações de saída. Assim, um pacote quando emitido numa entrada atravessa um conjunto de nós (*hops*) até a saída. Como uma árvore fornece caminhos únicos entre dois nós, é possível sempre encaminhar um pacote de dados de forma única até ao destino.

Um dos objetivos de uma rede de comunicações é minimizar a latência de comunicação, ou seja, minimizar o número de *switches* pelos quais um pacote tem de passar desde a origem até o destino. Numa rede como mostrada na Figura 1, a maior latência corresponde à maior distância entre nós terminais (medida pelo número de nós (*switches*)). Este valor corresponde ao diâmetro da árvore, que numa árvore binária completa é dado por  $2 * \log_2(N) + 2$ , em que  $N$  é o número de entradas e de saídas.

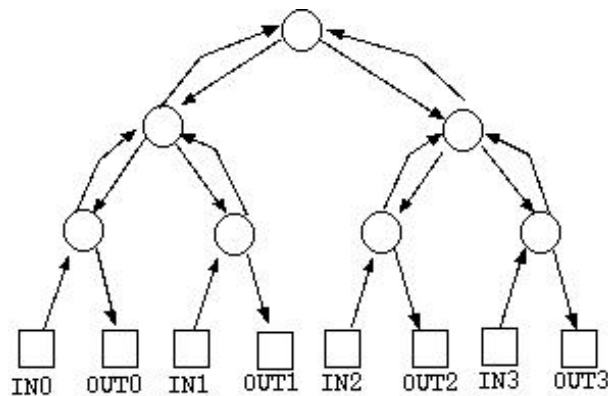


Figura 1: Exemplo de árvore binária completa.

## 2 Objetivo

O principal desafio deste trabalho é escrever um programa em que, dada uma configuração de uma **árvore binária de busca**, possivelmente não balanceada, determine o seu diâmetro e imprima a árvore em uma ordem determinada. Os dois exemplos mostrados na Figura 2 ilustram dois casos possíveis de árvores binárias de busca e os respectivos diâmetros.

No caso de árvores binárias, em geral, o diâmetro de uma árvore  $T$  é a maior das seguintes quantidades:

- o diâmetro da subárvore esquerda de  $T$ ;
- o diâmetro da subárvore direita de  $T$ ;

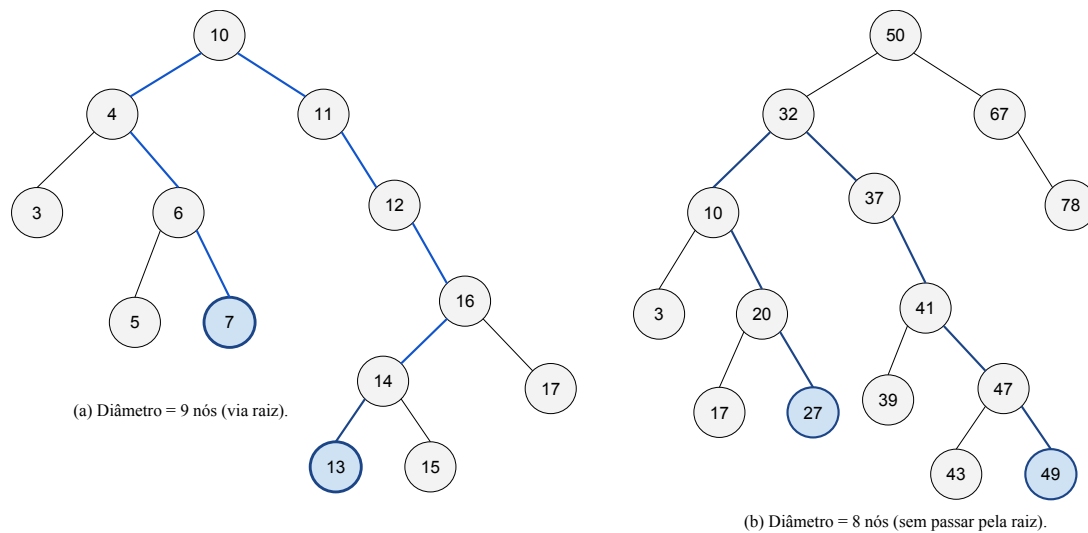


Figura 2: Exemplos de árvores binárias de busca e respectivos diâmetros.

- o maior caminho entre duas quaisquer folhas que passe pela raiz de  $T$  (este valor pode ser calculado a partir das profundidades das subárvores de  $T$ ).

### 3 Entrada

A primeira linha contém um inteiro positivo  $[N]$  e uma string  $[OR]$  que indicam o número de nós da árvore e o percurso que a árvore deve ser impressa (pré-, pós- ou in-ordem), respectivamente. O número de nós  $[N]$  tem valor menor ou igual a 100, enquanto que  $[OR]$  é uma string com valor in (para in-ordem), pos (para pós-ordem) ou pre (para pré-ordem).

A linha seguinte corresponde aos nós da árvore binária de busca, possivelmente a árvore está desbalanceada, sendo o número de nós superior ou igual a 1. Cada nó da árvore é caracterizado por um inteiro positivo qualquer **sem repetição**. Os valores dos nós estão separados por um espaço. O método de inserção em árvore binária de busca requer  $O(\log_2 N)$  vezes para o caso médio e necessita de  $O(N)$  no pior caso. A fim de introduzir um nó novo na árvore, seu valor é primeiro comparado com o valor da raiz. Se seu valor for menor que a raiz, é comparado então com o valor do filho da esquerda da raiz. Se seu valor for maior, está comparado com o filho da direita da raiz. Este processo continua até que o nó novo esteja comparado com um nó da folha, e então adiciona-se o filho da direita ou esquerda, dependendo de seu valor.

A execução termina quando os identificadores  $[N]$  e  $[OR]$  forem iguais a 0.

### 4 Saída

O seu programa deve produzir duas linhas de saída para cada caso. A primeira linha deve imprimir uma string com a frase “Diametro da arvore binaria:  $[D]$ ”, em que  $[D]$  é o valor calculado para o diâmetro da árvore correspondente. A segunda linha corresponde ao percurso na ordem determinada da árvore de entrada.

### 5 Exemplos

#### Exemplo 1

##### Entrada

```
13 in
10 4 11 3 6 5 7 12 16 14 17 13 15
14 pos
```

50 32 67 10 37 78 3 20 41 17 27 39 47 49  
0 0

### Saída

Diametro da arvore binaria: 9  
3 4 5 6 7 10 11 12 13 14 15 16 17  
Diametro da arvore binaria: 8  
3 17 27 20 10 39 49 47 41 37 32 78 67 50

## Exemplo 2

### Entrada

32 pos  
43 35 69 7 13 52 93 17 90 29 25 74 98 72 11 75 18 56 59 31 60 67 36 63 82 54  
73 21 44 19 76 58  
14 pos  
93 9 89 23 70 26 35 36 86 24 50 40 19 49  
3 in  
82 11 32  
22 in  
86 51 53 87 97 71 57 61 2 44 93 65 82 94 75 28 3 91 38 33 85 83  
0 0

### Saída

Diametro da arvore binaria: 17  
11 19 21 18 25 31 29 17 13 7 36 35 44 54 58 63 67 60 59 56 52 73 72 76 82 75 74 90 98 93 69 43  
Diametro da arvore binaria: 11  
19 24 49 40 50 36 35 26 86 70 23 89 9 93  
Diametro da arvore binaria: 3  
11 32 82  
Diametro da arvore binaria: 11  
2 3 28 33 38 44 51 53 57 61 65 71 75 82 83 85 86 87 91 93 94 97

## 6 Observações

- O número máximo de submissões é 10;
- Para a realização dos testes do SuSy, a compilação se dará da seguinte forma:  
gcc -std=c99 -pedantic -Wall -o lab8 lab8.c;
- Erros de memória como acesso a memória invalida, vazamento, ou uso de memória não inicializada acarretarão em penalidades na nota do trabalho;
- Você deve incluir, no início do seu programa, o seu nome e seu RA;
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.