

## Exercício 09 – Checar Busca



### Contexto

Está chegando a época natalina!

Nesse contexto, Frederico resolveu chamar os outros arboristas para enfeitar as árvores do ICMC com bolas vermelhas, entretanto eles decidiram que havia uma regra, eles só poderiam colocar uma bola a direita de outra se ela fosse maior que esta outra, e a esquerda se ela fosse menor.

Assim, eles animados começaram a colocar as bolas nos galhos das árvores e quando a primeira árvore ficou pronta, Frederico se deu conta de que precisaria checar se a árvore seguia a regra que propuseram, assim eles passaram toda a tarde olhando bola por bola para ver se alguma ordem estava errada!

Você, vendo essa situação, precisa agora salvar o natal do instituto criando um algoritmo que cheque se as bolas de uma árvore seguem a regra dos arboristas ou não, só assim conseguirão enfeitar as árvores a tempo!

### Descrição

Utilize o TAD árvore binária que já construíram para o exercício 08 e adicione uma função `ab_checar_busca`, que receba uma árvore binária e retorne se esta é uma árvore de busca ou não. O programa receberá diversas inserções e no final deve imprimir se a árvore é de busca seguindo a especificação da seção Saída.

A seguir está uma interface reduzida para o TAD Árvore Binária.

```
(ab.h)

#ifndef _ARVORE_BINARIA_H
#define _ARVORE_BINARIA_H

#include "item.h"
#define ERRO -32000
#define FILHO_ESQ 0
#define FILHO_DIR 1

typedef struct arv_bin AB;

AB *ab_criar(void);
bool ab_inserir (AB *T, ITEM *item, int lado, int chave);
void ab_imprimir (AB *T);
void ab_apagar_arvore(AB **T);

int ab_estritamente_binaria (AB *T); /*Essa função devolve 0
se a Árvore é estritamente binária e 1 caso contrário*/
int ab_checar_busca (AB *T); // Essa função devolve 0 se a
Árvore é de busca e 1 caso contrário

#endif
```

## Entrada

A entrada é iniciada pelo número N de elementos na árvore.

Em seguida está a sequência de inserção destes elementos na árvore binária. Em cada linha o primeiro inteiro é o valor a ser inserido, o segundo é a posição a ser inserido (0 para esquerda e 1 para direita) e o terceiro é qual deve ser o pai deste valor. A raiz sempre será a primeira inserção sem importar o valor de posição ou pai.

Exemplo:

Exemplo de Entrada		
3		
2	0	0
1	0	2
3	1	2

(Neste caso 2 é a raiz, 1 está a esquerda de 2 e 3 está a direita de 2)

## Saída

A saída será “É DE BUSCA SIM!” caso a árvore seja de busca, e “NÃO!” caso a árvore não seja.

Exemplo de Saída
É DE BUSCA SIM!

## Observações:

- O exercício deve ser desenvolvido individualmente por cada aluno, sendo este responsável por decidir as melhores opções de implementação.
- Somente as bibliotecas `stdio.h`, `stdlib.h` e `stdbool.h` podem ser utilizadas.
- Deve ser enviado um arquivo zip contendo toda a solução para o exercício. É necessária uma Makefile para compilar e executar o programa, os arquivos de implementação e interface do TAD Item e TAD Árvore Binária e o arquivo `main.c`.