UNIVERSIDADE FEDERAL DA FRONTEIRA SUL - UFFS
CURSO: CiêNCIA DA COMPUTAÇÃO
PROFESSOR: DOGLAS ANDRÉ FINCO
TRABALHO 2 – ESTRUTURA DE DADOS II

Descrição:

Implementar (em linguagem C) uma Árvore RedBlack que possui como chave valores inteiros positivos não repetidos. A cada novo nó inserido devem ser mantidas as propriedades da árvore, de modo a fazer as rotações e recoloramento dos nodos, se necessário.

Instruções:

O programa possui um menu para interação com o usuário com três opções: (1) Para inserir um elemento. (2) Para listar os elementos inseridos. (0) Para finalizar o programa.

(1) Inserção de um novo elemento: Sempre que selecionar esta opção, o programa irá pedir o valor da chave a ser inserida, após inserir o valor, o usuário deverá apertar "ENTER", para que ela seja adicionada na árvore, se o usuário tentar inserir uma chave já existente, o programa alerta que o elemento já foi inserido e retorna ao menu de opções.

(2) Listagem dos valores: Sempre que selecionar esta opção, o programa irá mostrar o nivel, a chave, a cor e a chave do pai de todos os elementos da árvore.

(0) Sair: Sempre que selecionar esta opção, o programa encerra.

Observações: O trabalho pode ser feito em duplas, porém a nota será individual. Somente um integrante da dupla submete o trabalho via moodle num arquivo com o nome dos dois integrantes e com extensão .zip, contendo todos os arquivos de sua implementação. Exemplo de nome de arquivo: Fulano_Ciclano.zip. As demais regras para o trabalho são as que constam no plano de ensino.

Estruturas e função de inicialiação da árvore
(OBS: Se optarem por utilizarem outras estruturas ou modificarem as existentes, fiquem a vontade, desde que isso não afete o objetivo fim do programa).

```
typedef struct _nodo{
	int chave;
	int nivel;
	int cor;
	struct _nodo *esq;
	struct _nodo *dir;
	struct _nodo *pai;
} TpNodo;
typedef struct _arvore{
	TpNodo *raiz;
	TpNodo *sentinela;
} TpArvore;

TpArvore *inicializa(){//aloca memoria para inicializar a arvore
	TpArvore *arvore=(TpArvore *)malloc(sizeof(TpArvore));
	TpNodo *sentinela=(TpNodo *)malloc(sizeof(TpNodo));
	arvore->raiz= NULL;
	arvore->nivelMax = 0;
	arvore->sentinela = sentinela;
	arvore->sentinela->chave = -1;
```

```
	arvore->sentinela->cor = 1;
	arvore->sentinela->pai = NULL;
	arvore->sentinela->esq = NULL;
	arvore->sentinela->dir = NULL;
	return arvore;
}

int main(){
	TpArvore *arvore=(TpArvore*)malloc(sizeof(TpArvore));
	arvore=inicializa();
}
```