Отчет по лабораторной работе №23 по курсу ЯМП

Студент группы М8О-107Б-19 Носов Эмиль Лаймонасович № по списку 16

Контакты www, e-mail, icq, skype_____

Работа выполнена: 27 марта 2021г.

Преподаватель: доц. каф. 806 Сластушенский Ю.В.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____201 __ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Динамические структуры данных, обработка деревьев.

2. **Цель работы:** Составить программу на языке Си для построения и обработки дерева общего вида или упорядоченного двочиного дервеа, содержащего узлы типа float, int, char или enum. Основные функции работы с деревьями реализовать в виде универсальных процедур или функций. Обработку осуществлять при помощи меню.
3. **Задание** (*вариант № 17*)**:** Проверить, является ли двоичное дерево самоподобным (подобным своему отражению).
4. **Оборудование** (лабораторное):
   ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
   НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
   Другие устройства _____
   _____

   Оборудование ПЭВМ студента, если использовалось:
   Процессор: Ryzen 7 4600H с ОП 16 Гб, НМД 256 Гб. Монитор 1920x1080
   Другие устройства _____
   _____

5. **Программное обеспечение (лабораторное):**
   Операционная система семейства _____, наименование _____ версия _____
   интерпретатор команд _____ версия _____
   Система программирования _____ версия _____
   Редактор текстов _____ версия _____
   Утилиты операционной системы _____
   _____
   Прикладные системы и программы _____
   Местонахождение и имена файлов программ и данных_____
   _____

   Программное обеспечение ЭВМ студента, если использовалось:
   Операционная система семейства Windows, наименование Windows 10 версия 2004
   интерпретатор команд bash версия 5.0.17.
   Система программирования _____версия _____
   Редактор текстов _____ версия _____
   Утилиты операционной системы _____
   _____
   Прикладные системы и программы _____
   _____
   Местонахождение и имена файлов программ и данных на домашнем компьютере _____
   _____

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Идея заключается в том, что если дерево симметрично, то возможно обойти его совершая аналогичные действия, то есть, при возможности/невозможности движения влево/вправо в левом поддерево можно/нельзя совершить движение вправо/влево в правом поддереве.

7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

```
#include<stdio.h>
#include<stdlib.h>

typedef struct tree* Tree;

struct tree {
	struct tree* left;
	struct tree* right;
	int value;
};

void lRr(Tree t, int d){
	if(t!=NULL){
		lRr(t->left,d+1);
		printf("%*s%d\n",2*d,"-",t->value);
		lRr(t->right,d+1);
	}
}

void printtree(Tree t){
	lRr(t,0);
	printf("\n");
}

Tree add(int b, Tree it){
	if (it==NULL) {it=(Tree)malloc(sizeof(struct tree)); it->value=b; it->right=NULL; it-
>left=NULL;}
	else {
		if(b>=(it->value)) {it->right=add(b, it->right);}
		if(b<(it->value)) {it->left=add(b, it->left);}
	}
	return(it);
}

Tree findprev(int b, Tree it){
	Tree d=it;
	if(it!=NULL){
		while(it->value!=b){
			d=it;
			if(it->value<b) {
			it=it->right;
			}
			if(it->value>=b) {
			it=it->left;
			}
		}
	}
	return(d);
}

Tree down(Tree it){
	while(it->left!=NULL) it=it->left;
	return(it);
}

void del(int b, Tree *it){
	Tree that=*it, prev, d;
	prev=that;
	prev = findprev(b, prev);
```

```
	if (that->value==b){
		if(that->right!=NULL) {
			*it=that->right;
			d=down(*it);
			d->left=that->left;
		}
		else if(that->left!=NULL) *it=that->left;
		else if(that->left==NULL) *it=NULL;
	}
	else if (b>=prev->value) {
		that = prev->right;
		if (that->right!=NULL) {
			prev->right=that->right;
			d=down(prev->right);
			d->left=that->left;
		}
		else if(that->left!=NULL) prev->right=that->left;
		else if(that->left==NULL) prev->right=NULL;
	}
	else if (b<prev->value) {
		that = prev->left;
		if (that->right!=NULL) {
			prev->left=that->right;
			d=down(prev->left);
			d->left=that->left;
		}
		else if(that->left!=NULL) prev->left=that->left;
		else if(that->left==NULL) prev->left=NULL;
	}
	free(that);
}

int lrr(Tree tl, Tree tr, int b){
	if (tl!= NULL && tr!=NULL){
		b=lrr(tl->left,tr->right, b);
		b=lrr(tl->right, tr->left, b);}
	else if (!((tl!= NULL || tr==NULL)&&(tl== NULL || tr!=NULL))) b++;
	printf("\n%d---------%d-----%d-------------%d%d\n",(tl!= NULL || tr==NULL)&&(tl== NULL ||
tr!=NULL),tl!= NULL && tr!=NULL,b,tl!=NULL,tr!=NULL);
	return(b);
}

void option(Tree t){
	Tree tl=t->left, tr=t->right;
	if(lrr(tl,tr, 0)==0) printf("Tree is self-similar\n");
	else printf("Tree isn't self-similar\n");
}

int main(){
	Tree t=NULL;
	char a;
	int b;
	printf("\n p - print \n a - add \n d - delete \n o - option \n q - quit \n\n");
	while(1>0){
	if(a!='\n')printf("Enter command: ");
	scanf("%c",&a);
	if(a!='\n')printf("\n");
	switch(a){
		case 'a':
				printf("Enter value: ");
				scanf("%d",&b);
				t=add(b, t);
				printf("\n");
```

```
                        break;
                case 'd':
                        printf("Enter value: ");
                        scanf("%d",&b);
                        del(b, &t);
                        printf("\n");
                        break;
                case 'p':
                        printf("This is tree in lRr:\n");
                        printtree(t);
                        printf("\n");
                        break;
                case 'o':
                        option(t);
                        break;
                case 'q': return (0);
        }
}
}
```

Пункты 1-7 отчета составляются сторого до начала лабораторной работы.

Допущен к выполнению работы. **Подпись преподавателя** _____

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

p - print

a - add

d - delete

o - option

q - quit

Enter command: a

Enter value: 5

Enter command: a

Enter value: 3

Enter command: a

Enter value: 7

Enter command: a

Enter value: 2

Enter command: a

Enter value: 4

Enter command: a

Enter value:

```
6

Enter command: a

Enter value: 8

Enter command: p

This is tree in lRr:
   -2
 -3
   -4
-5
   -6
 -7
   -8


Enter command: o

Tree is self-similar
Enter command: d

Enter value: 5

Enter command: p

This is tree in lRr:
     -2
   -3
     -4
 -6
-7
 -8


Enter command: o

Tree isn't self-similar
Enter command: q
```

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы _____

_____

_____

_____

11. **Выводы**

В ходе работы я составил программу на языке Си для построения и обработки упорядоченного двочиного дерева, содержащего узлы типа int. Реализоваал проверку самоподоия дерева. Основные функции работы с деревьями были реализованы в виде универсальных процедур или функций.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

_____

_____

_____

_____

_____

Подпись студента _____