Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Филиал

«Минский радиотехнический колледж»

Учебная дисциплина «Структуры и алгоритмы обработки данных»

**Выполнил**: Тамкович Денис

группа 62492

Минск

2018

**Лабораторная работа № 13**

**Тема работы: «Реализация алгоритмов поиска кратчайших путей в графе»**

**1. Цель работы**

Формирование умений работы с нелинейной структурой данных «бинарное дерево».

**2. Задание**

Дана последовательность чисел. Построить бинарное дерево, содержащее эти числа.

// Дана последовательность чисел.

// Построить бинарное дерево, содержащее эти числа.

// Подготовил: Тамкович Денис, группа 62492

**uses** crt;

**Type**

TreeType=^zapic;

Zapic = **record**

inf: integer;

Left, Right: TreeType;

**End**;

**Var**

len: integer;

**Procedure** Create\_Tree(tree:TreeType; n:integer);

**Var**

New\_tree:TreeType;

**Begin**

**if** n<=len **then**

**begin**

{переход на левый узел}

**New**(tree^.left);

New\_tree:=tree^.left;

New\_tree^.inf:=random(10);

Create\_Tree(New\_tree, n+1);

{переход на правый узел}

**New**(tree^.right);

New\_tree:=tree^.right;

New\_tree^.inf:=random(10);

Create\_Tree(New\_tree, n+1);

**end**;

**End**;

**Procedure** View\_tree(tree:TreeType; x, n:integer);

**begin**

gotoxy(x,2\*n);

write(tree^.inf);

**if** n<=len **then**

**begin**

//переход на левый узел

View\_tree(tree^.left,x-(20 **div** n),n+1);

//переход на првый узел

View\_tree(tree^.right,x+(20 **div** n),n+1);

**end**;

**end**;

**Var**

tree,root\_tree: TreeType;

**Begin**

randomize;

clrscr;

read(len);

//Создаётся и заполняется числовым значением корневой элемент дерева

**new**(tree);

tree^.inf:=random(10);

//запоминается адрес первого корневого элемента

root\_tree:=tree;

//создаётся дерево

Create\_Tree(tree,1);

//повторный проход дерева

View\_Tree(root\_tree,40,1);

**End**.

**4. Контрольные вопросы и задания**

*1.С чем связана популярность использования деревьев в программировании?*

Популярность деревьев связано с их простотой в реализации и возможностью создания иерархической структурой.

*2. Можно ли список отнести к деревьям?*

Список нельзя отнести к деревьям, т.к. дерево, это иерархическая структура.

*3. Какие данные содержат адресные поля элемента**бинарного дерева?*

***Адресные поля*** имеют указатель на узел ***бинарного дерева***.

*4. Может ли**бинарное дерево быть строгим и неполным?*

Нет не может, т.к. строгое дерево должно быть полным.

*5. Может ли**бинарное дерево быть нестрогим и полным?*

Да может, т.к. дерево нестрогое.

*6. Каким может быть почти сбалансированное**бинарное дерево: полным, неполным, строгим, нестрогим?*

 Дерево называется почти сбалансированным, если длины всевозможных путей от корня к внешним вершинам отличаются не более, чем на единицу.

*7. Чем отличаются, с точки зрения реализации алгоритма, прямой, симметричный и**обратный обходы**бинарного дерева?*

Для прямого обхода используется алгоритм TLR. Он включает следующие шаги:

1) просмотр корня T;

2) просмотр с помощью алгоритма TLR левого поддерева;

3) просмотр с помощью алгоритма TLR правого поддерева.

Обход в обратном порядке реализуется с помощью алгоритма LRT:

1) обход левого поддерева с помощью алгоритма LRT;

2) обход правого поддерева с помощью алгоритма LRT;

3) просмотр корня T.

Симметричный обход реализуется с помощью алгоритма LTR:

1) просмотр левого поддерева с помощью алгоритма LTR;

2) просмотр корня T;

3) просмотр правого поддерева с помощью алгоритма LTR.