Отчет по лабораторной работе №23 по курсу практикум на ЭВМ

Студент группы М8О-106Б-21, Деревянко Е.А. № по списку 6

Контакты www, e-mail, icq, skype derevankok9@gmail.com

Работа выполнена: «15» мая_____2022__г.

Преподаватель: ст.преподаватель каф. 806 Дубинин А.В.

Входной контроль знаний с оценкой _____

Отчет сдан «30» мая_____2022__ г., итоговая оценка _4___

Подпись преподавателя _____

1. Тема: Динамические структуры данных. Обработка деревьев.

2 **Цель работы:** Написать на Си динамическую структуру данных. Обработать дерево.

3 **Задание** (*вариант № 35*)**:** Составить программу на Си для построения и обработки двоичного дерева поиска, содержащего узлы типа enum. Основные функции работы с деревьями реализовать в виде универсальных процедур или функций, его обработка должна производиться в режиме текстового меню со следующими действиями: добавление нового узла, текстовая визуализация дерева, удаление узла, определить уровень двоичного дерева, на котором находится максимальное число вершин.

4 **Оборудование** (лабораторное):
ЭВМ _____, процессор _____, имя узла сети_____с ОП
НМД _____ . Терминал _____ адрес_____ Принтер
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:
Процессор _____ с ОП _____, НМД _____. Монитор
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 _____, 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 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)
Определим структуру дерева. В структуре содержатся:
- значение узла типа enum;
- ссылка на структуру узла (правое поддерево);
- ссылка на структуру узла (левое поддерево);
Зададим следующие функции:
1. node *add(node *tree, value_type value) — добавление узла в бинарное дерево поиска. Если узел пуст, то создаем ссылку новый узел и присваиваем ее структуре узла. Значению узла присвоим value, левому и правому поддеревьям — NULL;
2. node *minimal_node(node *tree) — рекурсивный поиск минимального (самого левого) узла поддерева. Нужна только для функции delete;
3. node *delete(node *tree, value_type value) — удаление узла из bst. Если дерево пусто, то не удаляем, дальше рекурсивно ищем узел в дереве по значению и проверяем: если у узла нет детей, то просто удаляем узел; если есть только левое (правое) поддерево, то удаляем узел и заменяем его на «корень» левого (правого) поддерева; если же есть 2 ребенка, то ищем минимальный узел правого поддерева и заменяем им удаляемый узел, затем запускаем функцию для удаления этого «минимального» узла.
4. size_t depth(node *tree) — глубина дерева. У пустого дерева глубина 0; у корня глубина 1. Вызовем рекурсивный подсчет глубины каждой стороны (левой и правой) в 2 переменные и выведем большую;
5. size_t width(node *tree, size_t level) — ширина дерева. Находится по принципу обхода каждой стороны, запоминания при этом уровня и вычисление количество вершин на каждом уровне с левой и с правой стороны, а потом сложение этих чисел;
6. size_t task(node *tree) — пока не дойдем до глубины дерева, будем подсчитывать ширину на каждом шаге и запоминать максимум;
7. node *destroy(node *tree) — уничтожить дерево. Если дерево не пустое, очищаем память у всех листьев и присваиваем им NULL; затем рекурсивно до корня;
8. value_type str_to_enum(char *elem) — перевод строки в элементы enum, с помощью strcmp;
9. void print_colors(value_type elem) — обратный перевод enum в строку. Обе функции основаны на ветвлении;
10. char *gets_(char *s) — функция, которая читает строку до символа «\n» или EOF. Нужна для реализации более понятного интерфейса;

7 **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].
1. Реализовать дерево, в соответствии с функциональной спецификацией;
2. Реализовать функцию, выполняющую задание по номеру варианта;
3. Написать интерфейс для взаимодействия с пользователем;

Тесты:

katerina@katerina-VivoBook-ASUSLaptop-X521EA-S533EA:~/newlabs/lab23/23_enum$ ./main

Write:
ADD to add elem in tree
DELETE to delete elem from tree
PRINT to show tree
TASK to print tree level with maximal width
COLORS to show enum colors list
HELP to see commands again
BYE to exit

ADD
Write color from enum to add it to tree:
RED
ADD
Write color from enum to add it to tree:
BLACK
ADD
Write color from enum to add it to tree:
YELLOW
PRINR
Wrong command
PRINT

RED
BLACK
YELLOW
TASK
Tree level with maximal width: 2

DELETE
Write color from enum to delete it from tree:
RED
PRINT

YELLOW
BLACK
TASK
Tree level with maximal width: 1
BYE

Пункты 1-7 отчета составляются строго до начала лабораторной работы.
Допущен к выполнению работы. **Подпись преподавателя _____**

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

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

Для более оптимизированного хранения и поиска элементов массива существуют такие динамические структуры, как двоичные деревья поиска. В ЯП СИ такие структуры можно реализовать, используя указатели и динамическое выделение памяти. В данной ЛР мы получили такое дерево, в котором можно хранить не только числа, а в том числе и элементы перечислимого типа.

Недочёты при выполнении задания могут быть устранены следующим образом:

_____

_____

_____

Подпись студента _____