



Отчёт по лабораторной работе № 23 по курсу Вычислительные системы

Студент группы М80-106Б Стрыгин Денис Дмитриевич, № по списку 22

Контакты www, e-mail, icq, skype

Работа выполнена: « » 2020 г.

Преподаватель: Ст. преп. каф.806 Дубинин А. В.

Входной контроль знаний с оценкой

Отчёт сдан « » марта 2020 г., итоговая оценка

Подпись преподавателя

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

2. **Цель работы:** Составить программу на языке С для построения и обработки дерева общего вида или упорядоченного двоичного, содержащего узлы типа enum. Основные функции работы с деревьями реализовать в виде универсальных процедур или функций. Обработка дерева должна происходить в режиме текстового меню со следующими действиями: добавление нового узла, визуализация дерева, удаления узла, вычисление функции.

3. **Задание (вариант №):** Определить уровень двоичного дерева, на котором находится максимальное число вершин.

4. **Оборудование(лабораторное):**

ЭВМ , процессор , имя узла сети с ОП Мб,
НМД Мб. Терминал адрес . Принтер
Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор с ОП Мб, НМД Мб. 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)

Для обработки команд построим конечный автомат (функции `read_cmd`), который вычленяет из входного потока строку команды и значение аргумента, попутно отслеживая ошибки и конец ввода.

Функция `void interactive_loop(treeptr *root? char *names[])` повторяет считывание команды и аргументов, пока не встретится EOF, после считывания функцией `read_cmd`, производится анализ данных об ошибках (соответствующий вывод при их наличии), при их отсутствии производится выполнение соответствующей функции.

Для нахождения уровня с макс количеством узлов напомним функцию `int find_width_level(treeptr root)`, которая с помощью функции `bool fill_width_vector *v`, заполняет вектор ширины уровней дерева. Затем находим максимальное значение в векторе и индекс макс значения, который и является номером искомого уровня.

При этом так как в узлах находится `enum`, то при добавлении узла вводится строка, в соответствии которой ставится одно из значений перечисления `words`, для этого напомним функцию `words_string_to_words (char *s char names[])`, которая возвращает вычисленное значение для строки `s`, в соответствии с заданным массивом слов `names`, который однозначно соответствует `enum words`, что упрощает вывод дерева, ведь если в узле хранится значение `val`, то ему соответствует строка `names[val]`

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

1. Написание структуры и основных функций бинарного дерева
2. Отладка функций
3. Написание функции преобразования строки в значение `enum`
4. Написание интерактивного ввода и его отладка
5. Написание функций обёрток для печати дерева, добавления и удаления узлов, корректирование вывода ошибок
6. Написание функции задания и используемых ею функций, также их отладка

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

Основным преимуществом бинарного дерева является возможная высокая эффективность реализации основанных на алгоритмах поиска и сортировки. Двоичные деревья применяются для построения структур, таких как множества, мультимножества, ассоциативные массивы

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____