МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

|  |
| --- |
| КАФЕДРА (название кафедры) |

КУРСОВАЯ РАБОТА (ПРОЕКТ)   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. техн. наук, доц. |  |  |  | О.В. Павлов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| ИСПОЛЬЗОВАНИЕ ЗАДАННЫХ СТРУКТУР ДАННЫХ И АЛГОРИТМОВ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ |
| по дисциплине: Структуры и алгоритмы обработки данных |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | Z1431 | |  |  |  | Быстров М.Д. |
|  | номер группы | |  | подпись, дата |  | инициалы, фамилия |
| Студенческий билет № | |  | |  |  |  |

|  |  |
| --- | --- |
| Шифр ИНДО | 09.03.04 |

Санкт-Петербург 20\_\_

ЗАДАНИЕ

Цель курсового проектирования: изучение структур данных и алгоритмов их обработки, а также получение практических навыков их использования при разработке программ. Задача курсового проекта: разработка информационной системы для заданной предметной области с использованием заданных структур данных и алгоритмов. Тема курсового проекта: «Использование заданных структур данных и алгоритмов при разработке программного обеспечения информационной системы».

Согласно алгоритму распределения вариантов определена комбинация переменных частей задания:

Предметная область – «Регистрация постояльцев в гостинице»;

Метод хеширования – открытое;

Метод сортировки – слиянием;

Вид списка – циклический однонаправленный;

Метод обхода дерева – прямой;

Алгоритм поиска слова в тексте – Боуера и Мура (БМ).

ВВЕДЕНИЕ

Предметная область – «Регистрация постояльцев в гостинице». Программное решение, созданное для работы в данной предметной области, позволит автоматизировать документооборот гостиничного обслуживания.

Предметная область, охватывающая взаимодействие с клиентами гостиничного сервиса, является благодатной почвой для автоматизации, поскольку в ходе работы непрерывно генерируются потоки новых данных (регистрация постояльцев, заселения и выселения). Автоматизация этих процессов позволит существенно сократить время на обслуживание конечных клиентов предприятие, что положительно отразится на экономических показателях предприятия.

Используя преимущества электронного документооборота, такие как скорость, надежность, защита от ошибок персонала, аналитические возможности, отчетные механизмы, предприятие получает возможность значительно повысить эффективность своей работы. Чем продуманнее и качественнее спроектировано, реализовано и протестировано ПО, тем больше потенциальные выгоды от его внедрения и использования.

В пояснительной записке будет описан процесс создания подобного решения.

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

1. Хеш-таблица с открытым хешированием

Хеш-таблица – структура данных, основанная на использования для хранения данных хеш-функции.

Хеш-функция – это функция отображения множества значений из какого-либо множества А в множество Б. В структурах данных и криптографии используются хеш-функции, множество Б в реализации которых представляет из себя конечное множество значений фиксированной длины (в частном случае – множество целочисленных значений).

Хеш-таблица построена на свойстве детерминированности хеш-фунцкции, т.е. неизменности выходных данных для одних и тех же входных данных. Если с помощью выходного значения хеш-функции возможно быстро рассчитать адрес в оперативной памяти, по которому можно получить доступ к данным, то это возможно использовать для построения структуры данных, позволяющей эффективно производить поиск данных по ключу.

В общем виде хеш-таблица представляет из себя массив фиксированного размера, элементами которого являются хранимые данные. Индексом элемента массива является значение, полученное при вычислении хеш-функции по значению ключа. Ключ также входит в состав данных, хранящихся в элементе массива.

Хеш-таблица с открытым хешированием является частным случаем хеш-таблицы. Её особенность состоит в том, что при возникновении коллизий (когда для разных ключей рассчитывается один и тот же хеш) происходит разрешение с помощью связного списка, т.е. в каждом элементе массива таблицы содержится ссылка на другие элементы, также находящиеся в хеш-таблице, но вставленные с разрешением коллизии. В таком виде каждый элемент таблицы является «головой» связного списка.

При использовании хеш-функции с хорошим распределением вставка и удаление элемента в таблице будет происходить за О(1). При возникновении большого количества коллизий эффективность будет снижаться в худшую сторону вплоть до О(N).

1. Односвязный циклический список

Связный список – структура данных, представляющая из себя набор элементов, связанных между собой ссылками.

Односвязный список – такая разновидность связного списка, в котором каждый элемент содержит ссылку на последующий элемент (если он имеется).

Циклический список – такой список, в котором первый и последний элемент связаны ссылками. В случае с односвязным списком это означает, что в последнем элементе будет содержаться ссылка на первый. В том случае, если в списке всего один элемент, он будет содержать ссылку на самого себя.

Типичная программная реализация списка представляет из себя объект-оболочку, содержащий в себе ссылки на первый и последний элемент списка и позволяющий производить вставку и удаление данных. Элементами списка являются ссылочные типы либо структуры, под которые память динамически выделяется в куче. Полями-ссылками на соседние объекты являются указатели (в других терминологиях - ссылки).

1. АВЛ-дерево поиска с прямым обходом

Дерево представляет из себя граф с однонаправленными связями, в котором из каждого узла может исходить не более двух связей. Узел, в который не приходит связей, является вершиной. Узлы, из которых не исходит связей, называются листьями.

Дерево поиска – структура данных, представляющая из себя такое дерево, которое состоит из узлов, способных сравниваться между собой и сохранять упорядоченность с помощью этого свойства. Чаще всего реализация принимает следующий вид: все элементы в правом поддереве вершины превосходят вершину по значениям, в левом – уступают. Идея структуры состоит в вставке, поиске и удалении элементов за логарифмическое время.

АВЛ-дерево – разновидность дерева поиска, при котором сохраняется сбалансированность по высоте: для каждой вершины разность между высотами левого и правого поддерева не превышает 1. При обнаружении факта утери баланса (при добавлении/удалении элементов) происходит балансировка дерева с помощью левых и правых поворотов, которые выполняются всегда за O(1). Это позволяет сохранить стабильное время работы со структурой при любом порядке вставки данных, к примеру при упорядоченной вставке значений от 1 до 100. При использовании обычного дерева поиска структура бы выродилась в односвязный список и поиск значения составлял бы O(N). При использовании АВЛ-дерева этот недостаток не будет проявлен.

1. Алгоритм Боуера и Мура

Алгоритм Боуера и Мура – модификация алгоритма прямого поиска фрагмента текста, основанная на использовании таблицы смещений. Если при прямом поиске после обнаружения несовпадения слово для поиска сдвигается на один символ вправо, при поиске Боуера и Мура слово сдвигается на количество символов, равное расстоянию от не совпадшего символа в тексте до конца слова поиска, если этот символ встречался в слове поиска. Если символ не встречался в слове поиска, слово поиска сдвигается на всю длину. Использование алгоритма позволяет уменьшить количество итераций, затрачиваемых на поиск фрагмента.

1. Сортировка слиянием

Сортировка слиянием – алгоритм сортировки, работающий за стабильное время *O(nlogn).* Принцип работы заключается в дроблении исходного массива данных на части до интервалов размером 1-2 элемента, их отдельной сортировке и итеративном слиянии получившихся массивов. Требует дополнительного объема памяти O(n).

ОПИСАНИЕ ПРОГРАММЫ