|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

**«Записи с вариантами, обработка таблиц»**

Вариант 2

Студент Пермякова Екатерина Дмитриевна

Группа ИУ7 – 32Б

Преподаватель Силантьева Александра Васильевна

2023 г.

# Условие задачи

Ввести список стран, содержащий

* название страны
* столицу
* *материк*
* необходимость наличия визы
* время полета до страны
* *минимальную стоимость отдыха*
* основной вид туризма
  + 1: экскурсионный - количество объектов, их основной вид (природа, история, искусство)
  + 2: пляжный – основной сезон, температура воздуха и воды
  + 3: *спортивный* – вид спорта (горные лыжи, серфинг, восхождения, сплав)

Вывести список стран на выбранном материке, со стоимостью ниже указанной, где можно заняться указанным видом спорта

Создать таблицу, содержащую не менее 40 записей с вариантной частью.

Произвести поиск информации по вариантному полю.

Упорядочить таблицу, по возрастанию ключей (где ключ – любое невариантное поле по выбору программиста), используя:

1. исходную таблицу
2. массив ключей

Используя 2 разных алгоритма сортировки (простой, ускоренный)

Оценить эффективность этих алгоритмов (по времени и по используемому объему памяти) при различной реализации программы, то есть, в случаях а) и б).

Обосновать выбор алгоритмов сортировки. Оценка эффективности должна быть относительной (в %).

# Техническое задание

**Исходные данные:**

Таблица из 40 стран, записанная в файле data\_table.txt

|  |
| --- |
| Возможности:  (таблица уже содержит 40 записей о странах)  1 - Произвести поиск информации по любому полю  2 - Добавить запись в конец таблицы  3 - Удаление записи по значению любого поля (После удаления таблица обновляется)  4 - Вывести исходную таблицу  5 - Вывести таблицу ключей  6 - Отсортировать таблицу ключей  7 - Вывести исходную таблицу в упорядоченном виде, используя упорядоченную таблицу ключей (выполняется сортировка таблицы ключей и обновление исходной таблицы)  8 - Отсортировать исходную таблицу  9 - Вывести список стран на выбранном материке, со стоимостью ниже указанной, где можно заняться указанным видом спорта  10 - Оценить эффективность  0 - Завершить программу  P S  Максимальный размер рабочей таблицы - 100 элементов  Таблица сортируется по возрастанию времени полета до страны  Оценка эффективности проводится с другой таблицей, большей по размеру  Вводить данные нужно только на английском |

Пользователю надо выбрать номер действия, которое он хочет совершить.

Если пользователь выберет действие 1 или 3, то ему нужно будет выбрать номер поля, по которому он хочет найти структуру и затем ввести ее. Для выбора номера поля выводится такое приглашение:

|  |
| --- |
| Выберете поле:  1 - название страны  2 - столицу  3 - материк  4 - необходимость наличия визы  5 - время полета до страны  6 - минимальную стоимость отдыха  основной вид туризма  --- экскурсионный  ------- 7 - количество объектов,  ------- 8 - их основной вид  --- пляжный  ------- 9 - основной сезон,  ------- 10 - температура воздуха,  ------- 11 - температура воды  --- спортивный  ------- 12 - вид спорта |

**Результат:**

Еще до взаимодействия с пользователем программа считывает данные о странах из файла в массив структур и создает таблицу ключей.

Описание программы в зависимости от выбора номера действия:

**Действие 1:**

Выводится приглашение к выбору номера поля, по которому пользователь будет искать страну. Вводится номер поля. Производится проверка правильности выбранного номера поля.

Вводится поле. Производится проверка правильности выбранного поля.

Производится поиск стран по введенному пользователем полю. Если найдено несколько подходящих стран, то выводятся на экран они все.

**Действие 2:**

Производится проверка что длина таблицы не превышает максимума и можно добавить еще один элемент.

Вводятся данные добавляемой страны. Производится проверка на правильность введенных данных.

Добавляется запись в конец таблицы (так же изменяется и файл с таблицей)

Обновляется таблица ключей

**Действие 3:**

Обновляется таблица внутри программы в соответствии с файлом. (Чтобы при удалении элемента правильно сместить индексы стран, на случай если таблица была до этого отсортирована)

Выводится приглашение к выбору номера поля, по которому пользователь будет искать страну, которую хочет удалить. Вводится номер поля. Производится проверка правильности выбранного номера поля.

Вводится поле. Производится проверка правильности выбранного поля.

Производится поиск стран по введенному пользователем полю. Если найдено несколько подходящих стран, то выводятся на экран они все и пользователю предлагается выбрать индекс страны, которую он хочет удалить.

Производится проверка правильности выбранного индекса.

Удаляется страна из таблицы со смещением индексов и перезаписывается файл.

Обновляется таблица ключей

**Действие 4:**

Выводится исходная таблица в том виде, в котором она хранится в программе

**Действие 5:**

Выводится таблица ключей в том виде, в котором она хранится в программе

**Действие 6:**

Сортируется таблица ключей, методом слияния и выводится на экран

**Действие 7:**

Обновляется таблица внутри программы (перезаписывается заново из файла), чтобы индексы исходной таблицы внутри массива совпадали с индексами внутри структуры и, следовательно, с индексами внутри таблицы ключей

Сортируется таблица ключей, методом слияния

Выводится исходная таблица в упорядоченном виде, используя упорядоченную таблицу ключей

**Действие 8:**

Сортируется исходная таблица, методом слияния и выводится на экран

**Действие 9:**

Вводятся материк, стоимость отдыха и вид спорта. Производится проверка на правильность ввода.

Выводится список стран на выбранном материке, со стоимостью ниже указанной, где можно заняться указанным видом спорта

**Действие 10:**

Считывается из отдельного файла большая таблица стран. И в дальнейшем идет работа с ней.

Производится оценка эффективности алгоритмов сортировки пузырьком и слиянием при использовании таблицы ключей и при сортировке сразу исходной таблицы. Оценка эффективности проводится по времени выполнения и по затраченному объёму памяти в процентном соотношении

Выводится таблица полученный данных

**Действие 0:**

Завершается программа

**Описание задачи, реализуемой программой:**

Производит действия, описанные в предыдущем пункте при работе со структурой, содержащей данные о странах

**Способ обращения к программе:**

После запуска программы ./app.exe пользователь выбирает номер действия, которое хочет выполнить (нумерация действий описана в программе) и далее взаимодействует с программой в зависимости от выбранного действия.

**Описание возможных аварийных ситуаций и ошибок пользователя:**

|  |  |
| --- | --- |
| **Ошибка** | **Вывод программы** |
| Неверный ввод названия страны | ERR\_NAME\_INPUT: ошибка ввода названия страны |
| Неверный ввод столицы | ERR\_CAPITAL\_INPUT: ошибка ввода столицы |
| Неверный ввод континента | ERR\_CONTINENT\_INPUT: ошибка ввода континента |
| Неверный ввод визы, если нужна вводится - 1, если нет - 0, другие значения не принимает | ERR\_VISE\_INPUT: если виза нужна вводится - 1, если нет - 0, другие значения не принимает |
| Неверный ввод времени полета до страны | ERR\_TIME\_INPUT: ошибка ввода времени полета до страны |
| Неверный ввод минимальной стоимости отдыха | ERR\_COST\_MIN\_INPUT: ошибка ввода минимальной стоимости отдыха |
| Неверный ввод вида туризма. Нужно ввести одно из чисел - (0, 1, 2) | ERR\_TYPE\_TOURISM\_INPUT: ошибка ввода вида туризма. Нужно ввести одно из чисел - (0, 1, 2) |
| Неверный ввод количества достопримечательностей в экскурсионном туризме | ERR\_COUNT\_SIGHTS\_INPUT: ошибка ввода количества достопримечательностей в экскурсионном туризме |
| Неверный ввод вида достопримечательностей | ERR\_SIGHTS\_INPUT: ошибка ввода вида достопримечательностей |
| Неверный ввод сезона | ERR\_SEASON\_INPUT: ошибка ввода сезона |
| Неверный ввод температуры воздуха в пляжном туризме | ERR\_AIR\_TEMPERATURE\_INPUT: ошибка ввода температуры воздуха в пляжном туризме |
| Неверный ввод температуры воды в пляжном туризме | ERR\_WATER\_TEMPERATURE\_INPUT: ошибка ввода температуры воды в пляжном туризме |
| Неверный ввод вида спорта | ERR\_TYPE\_SPORT\_INPUT: ошибка ввода вида спорта |
| Неверный ввод номера действия или поля | ERR\_NUM\_ACT\_INPUT: вы ввели неверный код поля или действия |
| Программа не может найти файл с данными об изначальной таблице | ERR\_FILE\_NOT\_EXIST: Файл с таким названием не существует |
| Файл с данными об изначальной таблице испорчен | ERR\_BAD\_FILE: файл неверного содержимого |
| Файл с данными об изначальной таблице слишком большой | ERR\_DYNAMIC\_MEMORY: ошибка выделения динамической памяти |
| Попытка удалить страну которой нет в таблице | ERR\_NO\_COUNTRY\_IN\_TABLE: попытка удалить страну которой нет в таблице |
| Записей в таблице слишком много | ERR\_TOO\_MANY\_COUNTIES: записей в таблице слишком много |
| Введен неверный индекс | ERR\_WRONG\_INDEX: введен неверный индекс |
| Попытка добавить запись в переполненную таблицу | ERR\_ADD: нельзя добавить запись: количество записей в таблице уже максимально |
| Попытка удалить последнюю запись | ERR\_DEL: нельзя удалить запись: в таблице должна быть хотя бы одна запись |

# Описание внутренних структур данных

Данные о стране записаны в структуре:

|  |
| --- |
| #define NAME\_LEN 30  #define CAPITAL\_LEN 30  #define CONTINENT\_LEN 30  #define TYPE\_SIGHTS\_LEN 30  #define SEASON\_LEN 30  #define SPORT\_LEN 30  struct country\_t  {  size\_t ind;  char name[NAME\_LEN + 1];  char capital[CAPITAL\_LEN + 1];  char continent[CONTINENT\_LEN + 1];  bool visa;  int time;  int cost\_min;  type\_tourism\_t type\_tourism;  union tourism\_t {  struct excursion\_tourism\_t excursion\_tourism;  struct beach\_tourism\_t beach\_tourism;  char sport\_type[SPORT\_LEN + 1];  } tourism;  }; |

Где:

* ind – индекс страны внутри файла
* name – название страны
* capital – название столицы
* continent – материк
* visa – надобность визы
* time - время полета до страны
* cost\_min – минимальная стоимость отдыха
* type\_tourism – индекс основного вида туризма, который является перечисляемым типом

|  |
| --- |
| typedef enum type\_tourism\_enum { EXCURSION\_TOURISM = 0, BEACH\_TOURISM = 1, SPORT\_TOURISM = 2 } type\_tourism\_t; |

Все виды туризма записаны объединением. С целью уменьшения затрачиваемого объёма памяти под структуру. Так как в данном случае, страна может быть описана только одним видом туризма.

* excursion\_tourism – структура экскурсионного туризма

|  |
| --- |
| struct excursion\_tourism\_t  {  int count\_sights;  char type\_sights[TYPE\_SIGHTS\_LEN + 1];  }; |

Где:

* + count\_sights – Количество достопримечательностей
  + type\_sights – Вид достопримечательностей
* beach\_tourism – структура пляжного туризма

|  |
| --- |
| struct beach\_tourism\_t  {  char season[SEASON\_LEN + 1];  int air\_tempr;  int water\_tempr;  }; |

Где:

* + season – сезон
  + air\_tempr – температура воздуха
  + water\_tempr – температура воды
* sport\_type – вид спорта

Таблица ключей записана как структура:

|  |
| --- |
| struct key\_table\_t  {  size\_t ind\_orig;  int time;  }; |

Где:

* ind\_orig – индекс страны в файле (в таблице)
* time - время полета до страны

Таблица стран хранится в файле и для работы с ней в начале работы программы записывается в массив структур стран.

# Алгоритм

**Алгоритм ввода номера действия:**

Выводится приглашение к вводу номера действия. При неверном выводе предлагается ввести снова.

**Алгоритм ввода номера поля страны:**

Такой же как алгоритм ввода номера действия

**Алгоритм поиска страны по полю:**

По введенному полю создается временная структура страны, у которой заполнено только искомое поле, определяется функция сравнения во введенному полю структур стран.

Создается массив индексов подходящих стран (массив целых чисел int)

Программа проходится по всему массиву структур стран и записывает индексы подходящих стран в массив индексов. В данном случает под индексом понимается индекс страны в массиве стран.

Выводится таблица стран имеющих индексы как в массиве индексов.

**Алгоритм добавления страны в конец таблицы:**

Производится проверка что длина таблицы не превышает максимума и можно добавить еще один элемент.

Вводятся данные добавляемой страны. Производится проверка на правильность введенных данных.

Данные о стране записываются в конец файла.

Структура страны добавляется в конец таблицы

**Алгоритм удаления страны из таблицы:**

Производится проверка что длина таблицы не превышает максимума и можно добавить еще один элемент.

Вводятся данные добавляемой страны. Производится проверка на правильность введенных данных.

Выбирается страна по алгоритму поиска страны по полю, только в случае нахождения нескольких подходящих стран, выводятся на экран они все и пользователю предлагается выбрать индекс страны, которую он хочет удалить. Производится проверка правильности выбранного индекса.

Из таблицы удаляется страна с выбранным индексом. Сдвигаются страны, находящиеся правее в таблице и уменьшаются их индексы.

Перезаписывается файл. В него записывается таблица хранящаяся в программе

**Алгоритм вывода упорядоченной исходной таблицы по таблице ключей:**

Обновляется таблица внутри программы (перезаписывается заново из файла), чтобы индексы исходной таблицы внутри массива совпадали с индексами внутри структуры и, следовательно, с индексами внутри таблицы ключей

Сортируется таблица ключей, методом слияния

Выводится исходная таблица в упорядоченном виде, данные о странах в исходной таблице выводятся в порядке индексов записанные в упорядоченной таблице ключей

**Алгоритм сортировки пузырьком:**

Классический без модификаций

**Алгоритм сортировки слиянием:**

Классический без модификаций

**Алгоритм замера времени сортировки:**

Из файла с большой таблицей считываются данные.

Перезаписывается время полета до страны внутри каждой структуры стран на рандомное.

Используется функция gettimeofday() – которая возвращает время в мс во время ее вызова.

До и после вызова каждой функции сортировки измеряется время с помощью функции gettimeofday(). Разница между временем после вызова функции сортировки и до и считается за время работы функции сортировки

**Тесты:**

Тесты ввода числа:

|  |  |  |
| --- | --- | --- |
| **Ввод** | **Вывод** | **Что проверяется** |
| 1  3  Asia | Список стран на азиатском континенте | Поиск стран на азиатском континенте |
| 2  Russia  Moscow  Eurasia  0  10  50000  2  skiing | Страна добавлена в конец таблицы | Добавить страну “ Russia”в конец таблицы |
| 3  2  Moscow | Страна удалена | Удалить страну со столицей “Moscow” |
| 4 | Исходная таблица | Вывести таблицу |
| 5 | Таблица ключей | Вывести таблицу ключей |
| 6 | Отсортированная таблица ключей | Отсортировать таблицу ключей |
| 7 | Упорядоченная исходная таблица | Вывести упорядоченную исходную таблицу по таблице ключей |

|  |  |  |
| --- | --- | --- |
| 8 | Упорядоченная исходная таблица | Отсортировать исходную таблицу |
| 9  Eurasia  900000  skiing | Список стран удовлетворяющий условиям  (Для начальной таблицы это Russia, France) | Вывести список стран на выбранном материке, со стоимостью ниже указанной, где  можно заняться указанным видом спорта |
| 10 | Таблица оценки эффективности  (ее можно посмотреть в пункте “Эффективность” в отчете) | Произвести оценку эффективности |

# Эффективность

**Длина таблицы: 12000**

|  |  |  |  |
| --- | --- | --- | --- |
| Сортировки | Пузырьком | Слиянием | Эффективность сортировки |
| Исходная таблица | 598 ms | 55 ms | 91% |
| Таблица ключей | 215 ms | 5 ms | 98% |
| Эффективность хранения | 64% | 91% |  |

Размера (в байтах):

|  |  |  |
| --- | --- | --- |
| Исходная таблица | Таблица ключей | На сколько процентов больше |
| 1920000 | 2112000 | 9% |

**Длина таблицы: 8000**

|  |  |  |  |
| --- | --- | --- | --- |
| Сортировки | Пузырьком | Слиянием | Эффективность сортировки |
| Исходная таблица | 261 ms | 36 ms | 86% |
| Таблица ключей | 91 ms | 3 ms | 97% |
| Эффективность хранения | 65% | 92% |  |

Размера (в байтах):

|  |  |  |
| --- | --- | --- |
| Исходная таблица | Таблица ключей | На сколько процентов больше |
| 1280000 | 1408000 | 9% |

**Длина таблицы: 4000**

|  |  |  |  |
| --- | --- | --- | --- |
| Сортировки | Пузырьком | Слиянием | Эффективность сортировки |
| Исходная таблица | 67 ms | 18 ms | 73% |
| Таблица ключей | 22 ms | 1 ms | 95% |
| Эффективность хранения | 67% | 94% |  |

Размера (в байтах):

|  |  |  |
| --- | --- | --- |
| Исходная таблица | Таблица ключей | На сколько процентов больше |
| 640000 | 704000 | 9% |

Работая с таблицей колючей, мы затрачиваем всего на 9% больше памяти, но выигрываем на 65-95% по времени.

Так же можно заметить что сортировка слиянием эффективнее сортировки пузырьком на 75-98%.

В итоге, используя более эффективную сортировку слиянием, вместо сортировки пузырьком, и используя таблицу ключей, мы проигрывая по памяти на 9%, выигрываем во времени. И сортировка таблицы стран вместо 598мс (сортировка пузырьком исходной таблицы из 1200 элементов) начинает работать за 5 мс (сортировка слиянием таблицы ключей из 1200 элементов) – это выигрыш во времени практически в 120 раз.

# Вывод

Написана программа, которая работает с таблицей стран.

Произведено сравнение эффективности работы с исходной таблицей и с таблицей ключей.

# Ответы на вопросы

1. *Как выделяется память под вариантную часть записи?*

Выделяется блок памяти размером с самый большой элемент вариантной части и после в этот блок памяти записывается один из элементов вариантной части. Причем, компилятор не отслеживается какой из элементов (какой тип данных) был туда записан.

Например, если в этот блок памяти могут быть записаны типы int и double, то выделяется память под double. и если туда записать int и обратиться double, то значение вещественного числа будет неопределённо, так же работает и в обратную сторону.

1. *Что будет, если в вариантную часть ввести данные, несоответствующие описанным?*

Данные не проверяются и следить за содержимым должен программист.

Но если в вариантной части записаны только int и double, а переменной вариантной части попытаются присвоить строку, будет ошибка компиляции.

1. *Кто должен следить за правильностью выполнения операций с вариантной частью записи?*

Программист

1. *Что представляет собой таблица ключей, зачем она нужна?*

Таблица ключей – это таблица, состоящая из поля общей таблицы, по которому идет сортировка или поиск, и индекса элемента общей таблицы, которому это поле принадлежит.

Она нужна для сокращения времени работы с общей таблицей.

1. *В каких случаях эффективнее обрабатывать данные в самой таблице, а когда использовать таблицу ключей?*

Использовать таблицу ключей эффективнее, когда элемент общей таблицы занимает большое количество памяти и работа с ним (перестановка элементов и т. д.) занимает значительно большее количество времени чем работа с одним полем строки таблицы.

То есть, когда выигрыш во времени начинает превышать проигрыш по объёму занимаемых данных.

1. *Какие способы сортировки предпочтительнее для обработки таблиц и почему?*

Сортировка слиянием, быстрая сортировка или сортировка кучей. То есть те сортировки, у которых сложность O(n\*logn). Так как они работают быстрее.