



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»

---

## ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2 ПО ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ

### Записи с вариантами. Обработка таблиц

Студент \_\_\_\_\_ Палладий Е.И.

Группа \_\_\_\_\_ ИУ7-31Б

Название предприятия: НУК ИУ МГТУ им. Н. Э. Баумана

Студент	_____ Палладий Е.И.
Преподаватель	_____ Барышникова М.Ю.

2024 г.

## **СОДЕРЖАНИЕ**

<b>1</b>	<b>Описание условия задачи</b>	<b>2</b>
<b>2</b>	<b>Техническое задание</b>	<b>3</b>
2.1	Исходные данные . . . . .	3
2.2	Описание задачи, реализуемой программой . . . . .	4
2.3	Способ обращения к программе . . . . .	4
2.4	Описание аварийных случаев . . . . .	4
<b>3</b>	<b>Описание структуры данных</b>	<b>6</b>
<b>4</b>	<b>Описание алгоритма</b>	<b>9</b>
<b>5</b>	<b>Тестовые данные</b>	<b>10</b>
<b>6</b>	<b>Исследование</b>	<b>12</b>
<b>7</b>	<b>Ответы на вопросы</b>	<b>15</b>
	<b>ЗАКЛЮЧЕНИЕ</b>	<b>17</b>

## 1 Описание условия задачи

Создать таблицу, содержащую не менее 40-ка записей (тип – запись с вариантами (объединениями)). Упорядочить данные в ней по возрастанию ключей, двумя алгоритмами сортировки: пузырьёк и пузырьёк с флагом, сортируемое поле: **столица**, Сортировка производится двумя способами: а) Используя таблицу, б) Используя массив ключей. Возможность добавления и удаления записей в ручном режиме, просмотр таблицы, просмотр таблицы в порядке расположения таблицы ключей обязательна. Осуществить поиск информации по варианту. Произвести исследование эффективности алгоритмов сортировки при использовании структур и ключей

## **2 Техническое задание**

### **2.1 Исходные данные**

Список стран. В файле каждая страна должна быть разделена переносом строки, поля разделены пробелом. Информация о стране: название, столица, материк, необходимость наличия визы, время полета до страны, минимальная стоимость отдыха, основной вид туризма:

1. Экскурсионный:
  - 1.1. Количество объектов
  - 1.2. Основной вид объектов (природа, искусство, история)
2. пляжный:
  - 2.1. Основной сезон
  - 2.2. Температура воздуха и воды
3. Спортивный:
  - 3.1. Вид спорта (горные лыжи, серфинг, восхождения)

### **Меню программы:**

1. Вывести информацию о программе
2. Загрузить список стран из файла
3. Сохранить список стран в файл
4. Добавить страну в конец списка
5. Удалить страну из списка по названию
6. Вывести список стран
7. Вывести список ключей
8. Вывести список стран по по списку ключей
9. Вывести список стран на выбранном материке, где можно заняться указанным видом спорта, со стоимостью отдыха меньше указанной
10. Отсортировать список ключей по столице
11. Отсортировать список фильмов по столице
12. Произвести и вывести исследование

## 13. Выход

### 2.2 Описание задачи, реализуемой программой

Сохранение, добавление, удаление, вывод, сортировка стран и/или ключей как в файл, так и в консоль. Проведение замерного эксперимента и исследования об эффективности использования массива структур и массива ключей

### 2.3 Способ обращения к программе

Запускается через терминал: ./app.exe. Затем необходимо выбрать одну из предложенных опций в меню. При выборе опции №12 (Произвести и вывести исследование) для построения графиков зависимости времени выполнения сортировки от количества элементов необходимо из корневой директории прописать команду: `gnuplot research/src/*.sh`. После этого графики будут расположены в директории `research/plots`

### 2.4 Описание аварийных случаев

#### 1. Аварийные случаи при записи и чтении фильма

##### 1.1. Ошибки при вводе любого строкового поля

- a) Пустой ввод
- b) Переполнение буфера

##### 1.2. Ошибки при вводе поля: Информации о необходимости наличия визы

- a) Пустой ввод
- b) Символ
- c) Число, отличное от 0 или 1

##### 1.3. Ошибки при вводе полей: Время полета до страны, Минимальная стоимость отдыха, Количество объектов

- a) Пустой ввод
- b) Символ
- c) Отрицательное число

##### 1.4. Ошибки при вводе полей: Основной вид объектов, Вид спорта:

- a) Пустой ввод

- b) Символ
  - c) Число не из диапазона [1; 3];
- 1.5. Ошибки при вводе полей: Температура воздуха, Температура воды:
  - a) Пустой ввод
  - b) Символ
  - c) Число, выходящее за границы типа `short`
- 2. **Аварийный случай при добавлении страны в конец списка**
  - 2.1. Список уже содержит максимальное количество элементов
- 3. **Аварийный случай при удалении страны из списка**
  - 3.1. Страна отсутствует в списке
- 4. **Аварийный случай при работе с файлом**
  - 4.1. Файл отсутствует
  - 4.2. Нет прав доступа для работы с файлом

### 3 Описание структуры данных

Листинг 1. Структура для хранения фильма с вариативным полем

```
1 typedef struct
2 {
3     char name[MAX_COUNTRY_NAME_LENGTH];
4     char capital[MAX_CAPITAL_LENGTH];           // Название столицы
5     char mainland[MAX_MAINLAND_LENGTH];         // Название материка
6     bool visa;                                   // Потребность в визу
7
8     uint32_t flying_time;                       // Время полета в минутах
9     uint32_t min_vacation_price;                // Минимальная цена отдыха
10
11     type_of_tourism tourism;                    // Перечисление туризма
12     type_t type;                               // Объединение структур туризма
13 } country_t;
```

---

#### Объяснение полей:

Поле	Описание
name	Название страны
capital	Название столицы
mainland	Название материка
visa	потребность в визе
flying_time	Время полета
min_vacation_price	Минимальная стоимость отдыха
tourism	Вид туризма
type	Union туризма

## Листинг 2. Объединение видов туризма

```
1 typedef union
2 {
3     sightseeing_t sightseeing;           // Экскурсионный
4     beach_t beach;                      // пляжный
5     sport_t sport;                      // Спортивный
6 } type_t;
```

---

### Объяснение полей:

Поле	Описание
sightseeing	Структура с экскурсионным видом туризма
beach	Структура с пляжным видом туризма
sport	Структура со спортивным видом туризма

## Листинг 3. Структуры туризма

```
1 // Экскурсионный вид отдыха
2 typedef struct
3 {
4     uint32_t objects_amount;           // Количество объектов
5     type_of_objects objects_type;      // Вид объекта
6 } sightseeing_t;
7
8 // Пляжный вид отдыха
9 typedef struct
10 {
11     char season[MAX_SEASON_LENGTH];   // Сезон
12     short water_temperature;           // Температура воды
13     short air_temperature;             // Температура воздуха
14 } beach_t;
15
16 // Спортивный вид отдыха
17 typedef struct
18 {
```



```

19         type_of_sport sport_type;
20     } sport_t;

```

---

### Объяснение полей:

Поле	Описание
objects_amount	Количество объектов
objects_type	Вид объекта
season	Сезон
water_temperature	Температура воды
air_temperature	Температура воздуха
sport_type	Вид спорта

### Листинг 4. Возможные поля туризма

```

1     //Вид туризма
2     typedef enum
3     {
4         SIGHTSEEING = 1,        // Экскурсионный
5         BEACH,                  // Пляжный
6         SPORT                    // Спортивный
7     } type_of_tourism;
8
9     //Вид объектов
10    typedef enum
11    {
12        NATURE = 1,              // Природа
13        ART,                     // Искусство
14        HISTORY                  // История
15    } type_of_objects;
16
17    //Вид спорта
18    typedef enum
19    {
20        MOUNTAIN_SKIING = 1,     // Горные лыжи
21        SURFING,                 // Сёрфинг
22        CLIMBING                 // Скалолазание
23    } type_of_sport;

```

---

## 4 Описание алгоритма

1. Ввод пункта меню
2. Выбран пункт 1:
  - Информация о программе выводится на экран
3. Выбран пункт 2:
  - Вводится названия файла, из которого будет происходить запись
  - Список стран из файла загружается в массив структур
4. Выбран пункт 3:
  - Вводится названия файла, в который будет происходить запись
  - Список стран из массива структур загружается в файл
5. Выбран пункт 4:
  - Вводится страна из консоли
  - Страна добавляется в массив, если там есть место
6. Выбран пункт 5:
  - Вводится страна из консоли
  - Если страна есть в массиве - она удаляется
  - Длина массива уменьшается, если страна была удалена
7. Выбран пункт 6-9:
  - Список стран/ключей выводится в консоль
8. Выбран пункт 10-11:
  - Список фильмов или список ключей сортируется по столице
9. Выбран пункт 12:
  - Производится сортировка фильмов при разном количестве записей
  - Выводится статистика сортировки
  - При необходимости можно построить графики зависимости времени от количества фильмов

## 5 Тестовые данные

### Позитивные тесты

Тест	Входные данные	Выходные данные
Загрузить список стран из файла	data.txt	Данные успешно загружены
Сохранить список стран в файл	data.txt	Данные успешно сохранены
Добавить страну в конец списка	name capital mainland 1 20 30 1 12 1	Страна успешно добавлена
Вывести список стран		name capital mainland + 10 20 Sightseeing: 2 - Nature
Вывести список ключей		0 0 capital
Вывести список стран по списку ключей		name capital mainland + 10 20 Sightseeing: 2 - Nature
Вывести список стран по заданию	mainland 2 30	name capital mainland + 10 20 Sport: 2
Удалить страну из списка по названию	name	Страна успешно удалена
Отсортировать список ключей по столице		Список ключей успешно отсортирован
Отсортировать список фильмов по столице		Список фильмов успешно отсортирован

Произвести и вывести исследование		Исследование проведено успешно
Выход		Пока!

### Негативные тесты

Тест	Входные данные	Выходные данные
Пустой ввод строки	name capital _	EMPTY STRING ERROR!
Переполнение строки	name capital qwertyuiop qwertyui	BUF OVERFLOW ERROR!
Символ вместо числа	name capital mainland w 20 30 1 12 1	INPUT/OUTPUT ERROR!
Число выходит за диапазон short	name capital mainland й 20 12345678 1 12 1	RANGE ERROR!
Файл не существует (При загрузке данных)	qwerty.txt	ERR WITH FILE!
Переполнение массива	...	RANGE ERROR!
Страна для удаления отсутствует	mainland	ERR COUNTRY NOT FOUND
Список стран по заданию пустой	mainland 1 200	ERR COUNTRY NOT FOUND

## 6 Исследование

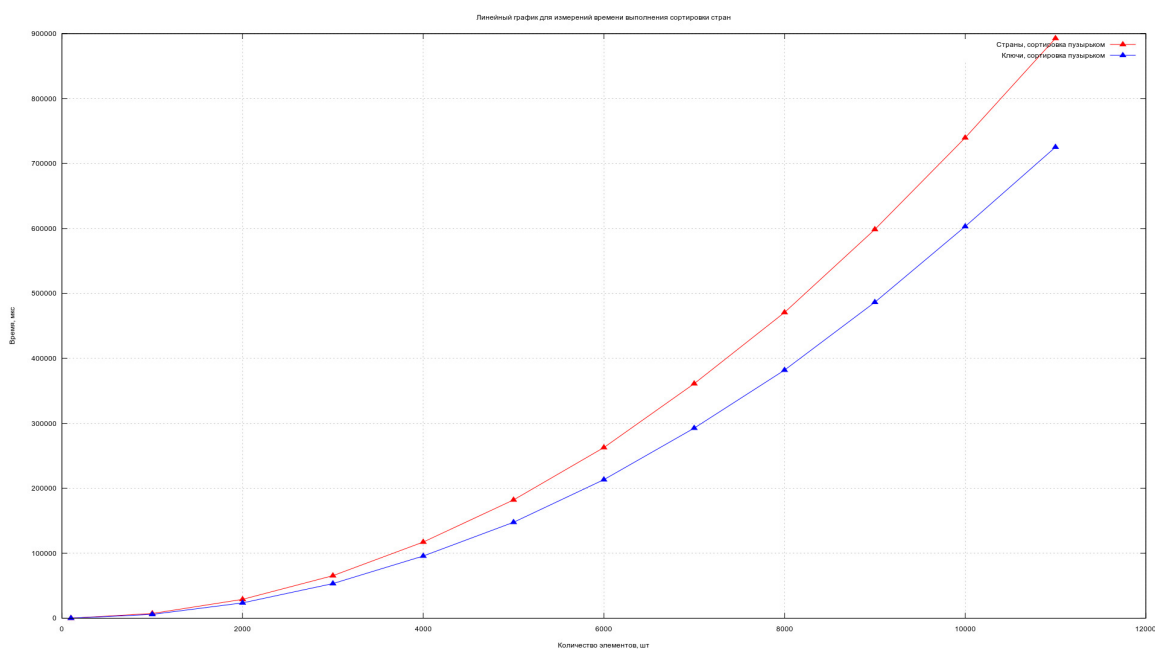


Рисунок 1. Зависимость времени между сортировкой стран и ключей

### Сортировка пузырьком. Таблица

Кол-во элементов	Время, мкс	Память, байт
1000	7295	80000
4000	117279	320000
80000	470726	640000
11000	892718	880000

### Сортировка пузырьком. Ключи

Кол-во элементов	Время, мкс	Память, байт
1000	5931	24000
4000	95769	96000
80000	381975	192000
11000	725277	264000

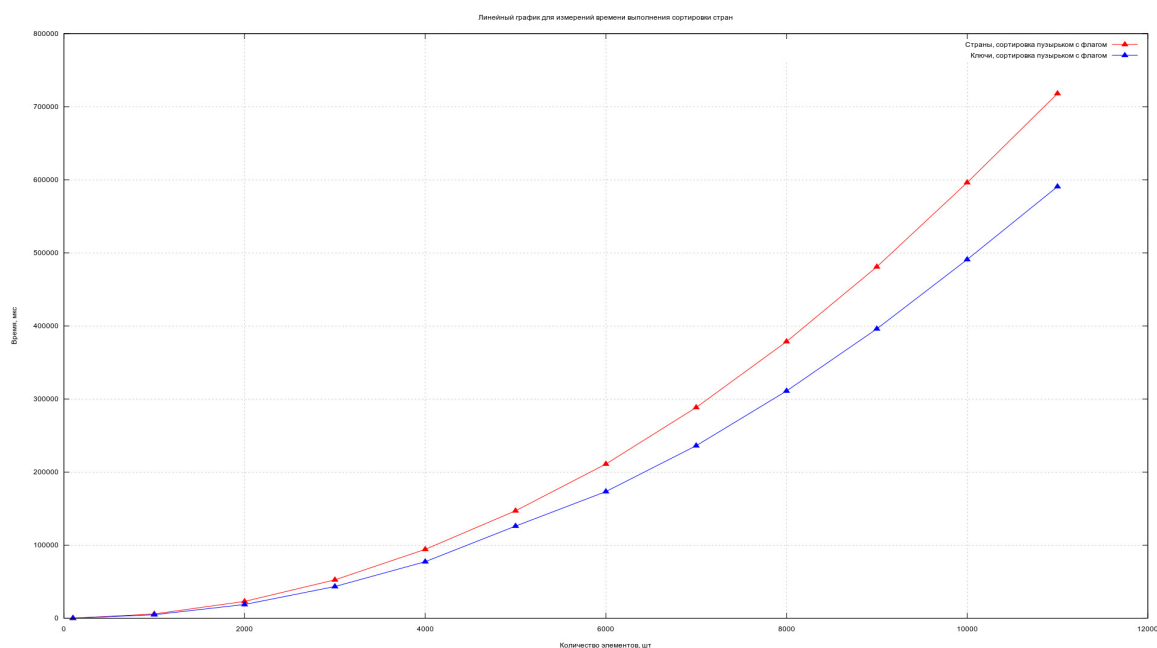


Рисунок 2. Зависимость времени между сортировкой с флагом стран и ключей

### Сортировка пузырьком с флагом. Таблица

Кол-во элементов	Время, мкс	Память, байт
1000	23028	80000
4000	94265	320000
80000	378553	640000
11000	717998	880000

### Сортировка пузырьком с флагом. Ключи

Кол-во элементов	Время, мкс	Память, байт
1000	4792	24000
4000	77359	96000
80000	310965	192000
11000	590672	264000

Как видно из полученных данных, использование ключей уменьшает употребление памяти примерно на 107%, а выигрыш по времени составляет примерно 20%

## Дополнительные графики:

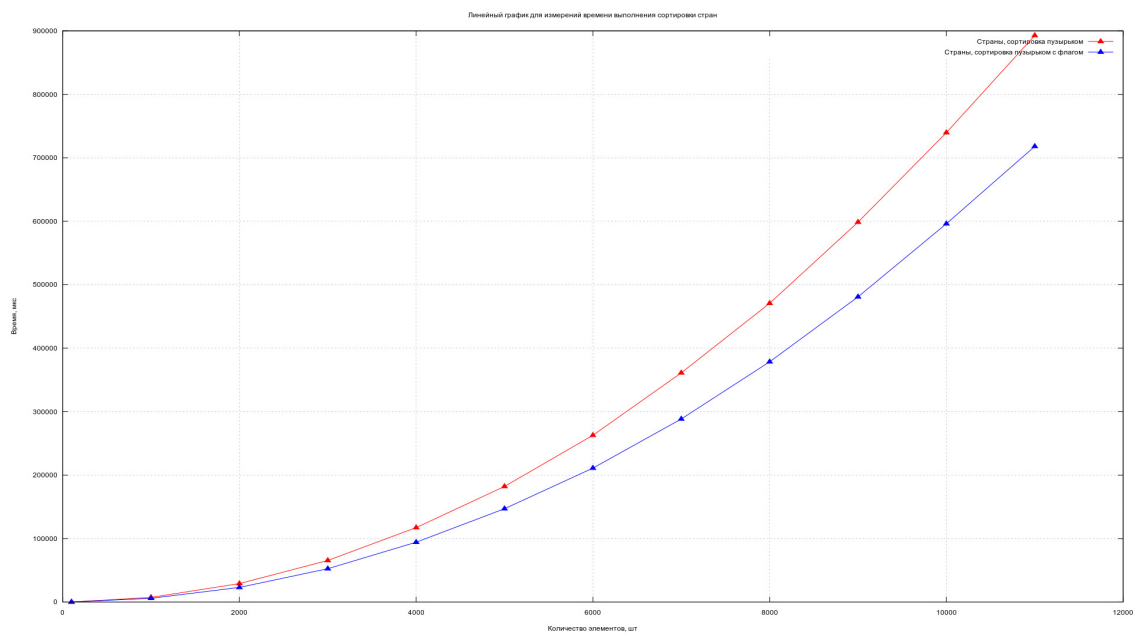


Рисунок 3. Зависимость времени между сортировкой стран

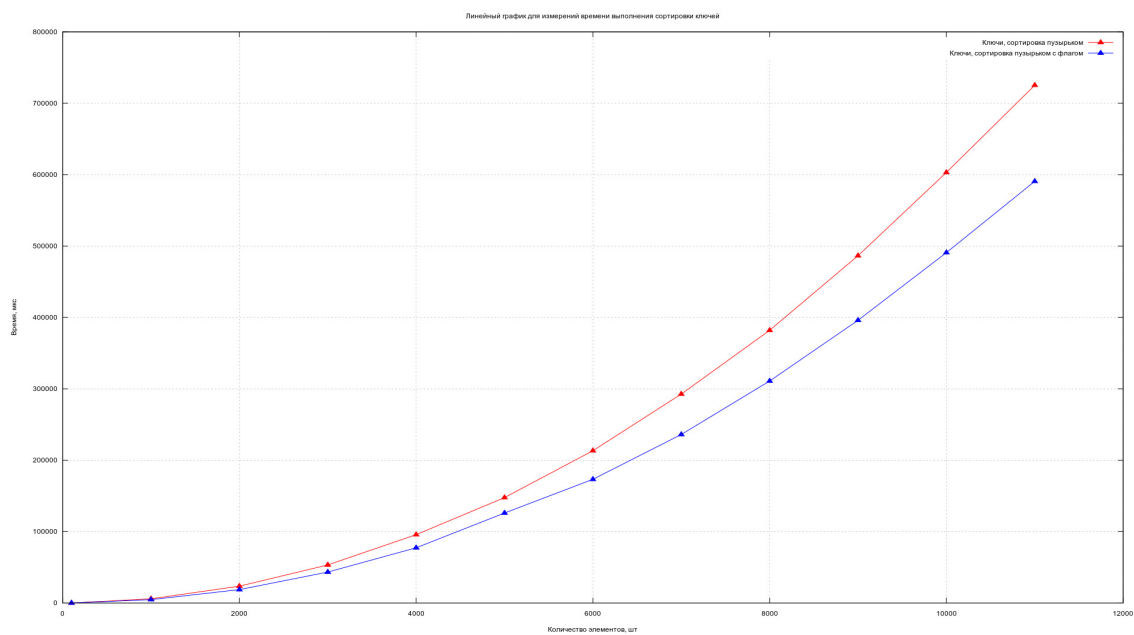


Рисунок 4. Зависимость времени между сортировкой ключей

## **7 Ответы на вопросы**

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

Для вариантной части записи используется объединение. Где каждое поле объединения - структура возможной записи. Такой формат обеспечивает минимизацию использования памяти.

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

Если такой ввод не вернет ошибку в программе, то дальнейший исход событий не предсказуем. Могут произойти разные ошибки, в том числе аварийное завершение программы.

### **3. Кто должен следить за правильностью выполнения операций с вариантной**

Только программист. Он должен обеспечить правильную запись вариантной части или вывод ошибки, при неверном вводе от пользователя.

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

Таблица ключей - структура данных, содержащая ключи или идентификаторы к исходной таблице. Таблица ключей занимает меньше памяти, поэтому выгодна в при частом использовании исходной таблицы.

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

Если при работе с исходной таблицей часто происходит обмен, сравнение, вставка, удаление объектов, то использование таблицы ключей поможет сократить время работы этих подпрограмм. Если происходит обращение к большому количеству разных полей таблицы, то обрабатывать данные в



таблице выгоднее

## 6. Какие способы сортировки предпочтительнее для обработки таблиц и почему?

Если данные в таблице почти отсортированы, то сортировка вставками подойдет лучше всего, так как будет осуществлено небольшое количество обменов

В общем случае, очевидно, лучше всего работают сортировки с временной сложностью  $O(N \log N)$ . Так как быстрее этой константы отсортировать физически невозможно. Однако есть всеми любимая **Сталинская сортировка** Которая, между прочим, работает за  $O(N)$ , но есть нюанс:)

Пример такой сортировки работающей за  $O(N \log N)$ : `Heap sort`.

## **ЗАКЛЮЧЕНИЕ**

Очень много информации хранятся в разных таблицах, многие из них имеют вариантные поля. В таких ситуациях программисту важно оценить, с какими операциями и данными ему придется иметь дело, чтобы подобрать лучшие по эффективности алгоритмы и методы обработки таблиц