Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №5**

**«ИССЛЕДОВАНИЕ АЛГОРИТМОВ СОРТИРОВКИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Носкова Анастасия Владимировна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы**

Получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

1. **Формулировка задания**

Вариант: 16

1. Реализовать сортировку данных с помощью алгоритма выбором.
2. Реализовать сортировку данных с помощью быстрого алгоритма
3. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виде передаваемой в подпрограмму функции).
4. Считывание и вывод данных необходимо производить из текстового файла.
5. Для демонстрации работы программных реализаций самостоятельно подготовить варианты входных данных (при этом объем текстовых файлов должен позволять оценить скорость работы программ).
6. **Описание алгоритма**

**Задание 1 – Алгоритм выбором**

1. Открывается файл "popit.txt" для чтения;
2. Считывается значение переменной "n" из файла;
3. Считываются элементы массива "a" из файла в цикле;
4. Закрывается файл "popit.txt";
5. Выводится пользователю сообщение о выборе порядка сортировки по возрастанию или убыванию;
6. Читается выбор пользователя в переменную "choice";
7. В зависимости от выбора пользователя вызывается процедура сортировки "sortselect" с соответствующим параметром "b" (True для возрастания и False для убывания);
8. Открывается файл "outpopit.txt" для записи;
9. Записываются отсортированные элементы массива "a" в файл в цикле;
10. Закрывается файл "outpopit.txt".

**Задание 2 – Быстрый алгоритм**

1. Открывается файл "z2input.txt" для чтения.
2. Считывается значение переменной n из файла.
3. Считывается массив a из файла.
4. Закрывается файл "z2input.txt".
5. Выводится на экран сообщение "Выберите порядок сортировки:".
6. Выводится на экран сообщения "1. По возрастанию" и "2. По убыванию".
7. Запрашивается у пользователя ввод номера порядка сортировки.
8. Считывается введенное пользователем значение в переменную choice.
9. В зависимости от значения переменной choice:
   1. Если choice равен 1, вызывается процедура min (a, 1, n), которая сортирует массив a по возрастанию.
   2. Если choice равен 2, вызывается процедура max (a, 1, n), которая сортирует массив a по убыванию.
   3. Если choice не равен ни 1, ни 2, выводится сообщение "Некорректный выбор." и завершается выполнение программы.
10. Открывается файл "z2output.txt" для записи.
11. Записывается отсортированный массив a в файл.
12. Закрывается файл "z2output.txt".
13. **Схема алгоритма**

**Задание 1**

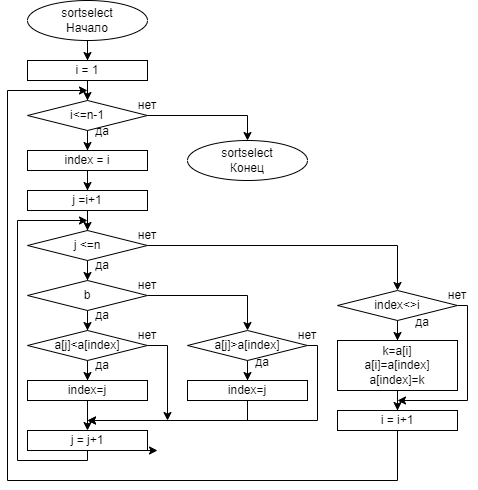


Рисунок 1 – схема процедуры sortselect– алгоритм выбором

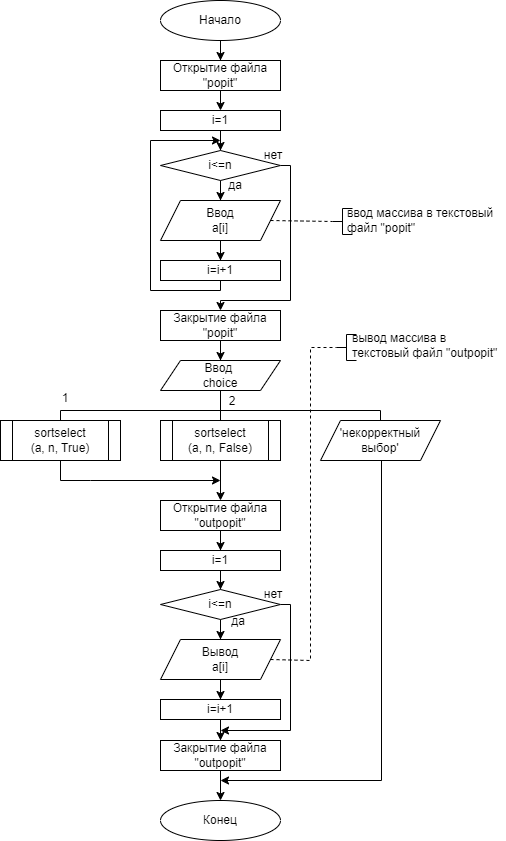


Рисунок 2 – схема алгоритма – алгоритм выбором

**Задание 2**

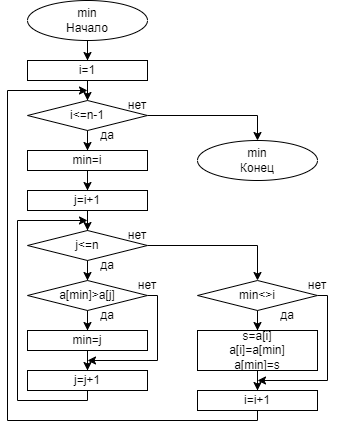


Рисунок 3 – схема процедуры min – быстрый алгоритм

Изображение выглядит как текст, снимок экрана, Шрифт, графический дизайн

Автоматически созданное описание

Рисунок 4 – схема процедуры max – быстрый алгоритм

Изображение выглядит как текст, снимок экрана, графический дизайн, Шрифт

Автоматически созданное описание

Рисунок 5 – схема алгоритма – быстрый алгоритм

1. **Код программы**

**Задание 1**

**type**

arr = **array**[1..100] **of** Integer;

**var**

f1, f2: text;

i, n: Integer;

a: arr;

**procedure** sortselect(**var** a: arr; n: Integer; b: Boolean);

**var**

i, j, index, k: Integer;

**begin**

**for** i := 1 **to** n - 1 **do begin**

index := i;

**for** j := i + 1 **to** n **do begin**

**if** b **then begin**

**if** a[j] < a[index] **then** index := j;

**end**

**else begin**

**if** a[j] > a[index] **then**

index := j;

**end**;

**end**;

**if** index <> i **then begin**

k := a[i];

a[i] := a[index];

a[index] := k;

**end**;

**end**;

**end**;

**begin**

assign(f1, 'popit.txt');

reset(f1);

readln(f1, n);

**for** i := 1 **to** n **do**

read(f1, a[i]);

close(f1);

writeln('Выберите порядок сортировки:');

writeln('1. По возрастанию');

writeln('2. По убыванию');

write('Введите номер: ');

**var** choice: Integer;

readln(choice);

**case** choice **of**

1: sortselect(a, n, True);

2: sortselect(a, n, False);

**else begin**

writeln('Некорректный выбор.');

Halt;

**end**;

**end**;

assign(f2, 'outpopit.txt');

rewrite(f2);

**for** i := 1 **to** n **do**

write(f2, a[i], ' ');

close(f2);

writeln('Результат сортировки записан в файл outpopit.txt!');

**end**.

**Задание 2**

**type**

mass = **array**[1..10000] **of** integer;

**var**

f1, f2: text;

i,n, z: integer;

a: mass;

**procedure** min(**var** a: mass; l, r: integer);

**var**

s, min, i, j: integer;

**begin**

**for** i := 1 **to** n - 1 **do**

**begin**

min := i;

**for** j := i + 1 **to** n **do**

**if** a[min] > a[j] **then**

min := j;

**if** min <> i **then**

**begin**

s := a[i];

a[i] := a[min];

a[min] := s;

**end**;

**end**;

**end**;

**procedure** max(**var** a: mass; l, r: integer);

**var**

s, max, i, j: integer;

**begin**

**for** i := 1 **to** n - 1 **do**

**begin**

max := i;

**for** j := i + 1 **to** n **do**

**if** a[max] < a[j] **then**

max := j;

**if** max <> i **then**

**begin**

s := a[i];

a[i] := a[max];

a[max] := s;

**end**;

**end**;

**end**;

**begin**

assign(f1, 'z2input.txt');

reset(f1);

readln(f1, n);

**for** i := 1 **to** n **do**

read(f1, a[i]);

close(f1);

writeln('Выберите порядок сортировки:');

writeln('1. По возрастанию');

writeln('2. По убыванию');

write('Введите номер: ');

**var** choice: Integer;

readln(choice);

**case** choice **of**

1: min(a, 1, n); // сортировка по возрастанию

2: max(a, 1, n); // сортировка по убыванию

**else begin**

writeln('Некорректный выбор.');

Halt;

**end**;

**end**;

assign(f2, 'z2output.txt');

rewrite(f2);

**for** i := 1 **to** n **do**

write(f2, a[i], ' ');

close(f2);

writeln('Результат сортировки записан в файл outpopit.txt!');

**end**.

1. **Результат выполнения программы:**

**Здание 1**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5  23 12 45 8 34 | 45 34 23 12 8 |
| 8  5 17 29 8 12 3 41 20 | 3 5 8 12 17 20 29 41 |

**Здание 2**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1  99 | 99 |
| 6  10 20 30 40 50 60 | 60 50 40 30 20 10 |

1. **Вывод**

Входе выполнения данной работы мы научились работать с двумя методами сортировки – методом быстрого алгоритма и алгоритма выбором.

Алгоритм выбора — это простой алгоритм сортировки, который сортирует массив элементов в порядке возрастания или убывания. Этодостигается путем обхода массива для поиска минимального элемента и замены его элементом с индексом 0 в отсортированном подмассиве. Затем несортированный подмассив снова просматривается, и минимальный элемент находится и переставляется в правильное положение.

Быстрый алгоритм также относится к алгоритмам «разделяй и властвуй». Его используют чаще других алгоритмов, описанных в этой статье. При правильной конфигурации он чрезвычайно эффективен и не требует дополнительной памяти, в отличие от сортировки слиянием. Массив разделяется на две части по разные стороны от опорного элемента. В процессе сортировки элементы меньше опорного помещаются перед ним, а равные или большие — позади.