**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(МИИТ)**

|  |
| --- |
| **ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Кафедра «Управление и защита информации»

**Лабораторная работа №8**

**«**Реализация алгоритма пирамидальной сортировки**»**

по дисциплине

**«Методы программирования»**

**Выполнил:** ст. гр. ТКИ-311,

Виноградова В.Л.

**Проверил:**

к.т.н., доц. Логинова Л.Н.,

к.т.н., доц. Сафронов А.И.

**Москва – 2022 г.**

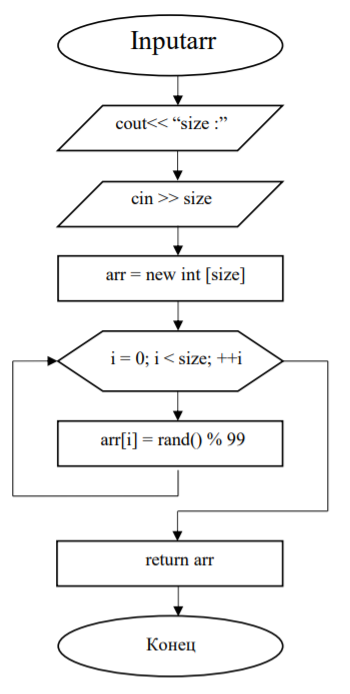
1. Цель работы

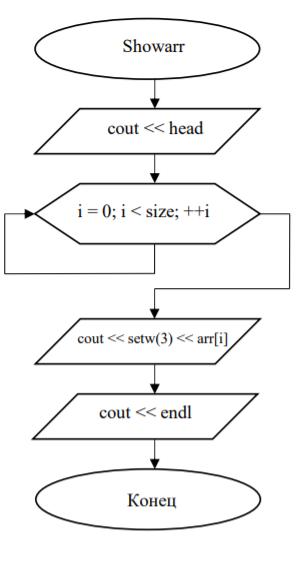
Изучить алгоритм пирамидальной сортировки.

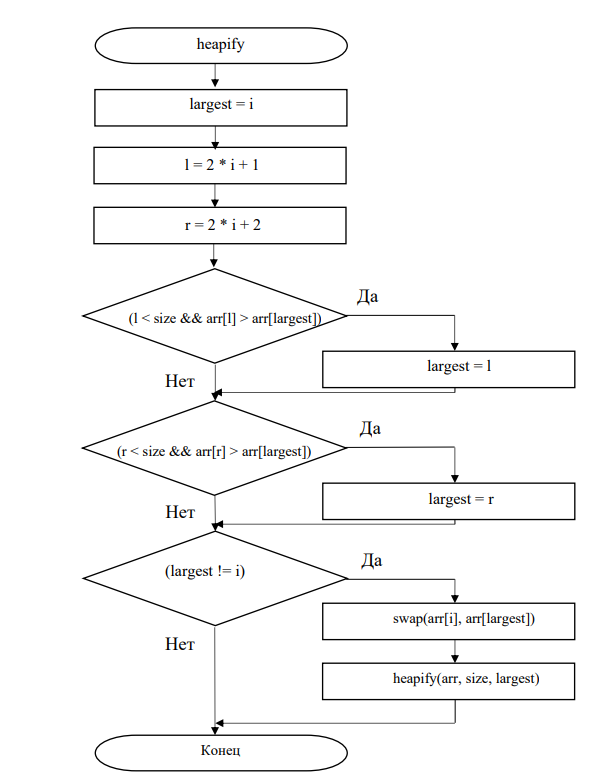
1. Постановка задачи

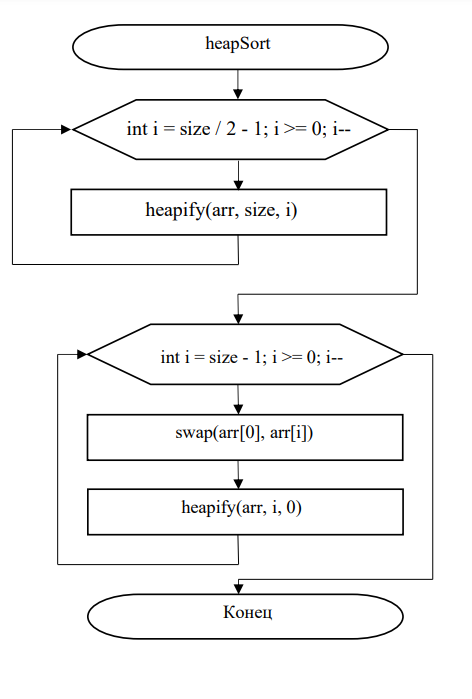
Реализовать алгоритм пирамидальной сортировки на любом языке программирования, ориентируясь на псевдокод. Сделать вывод о проделанной работе.

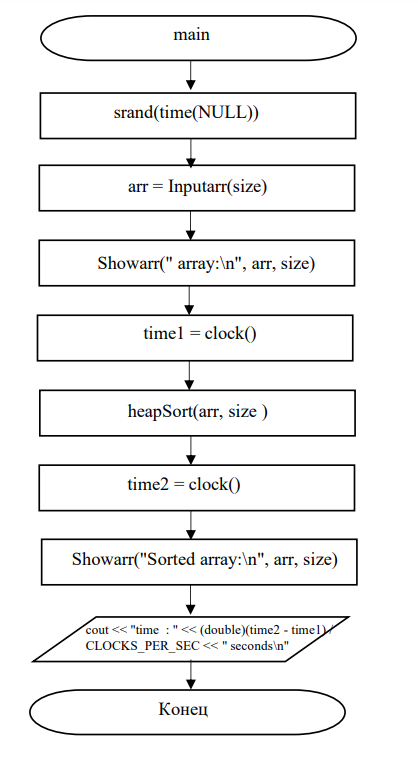
1. Блок-схема





****

****

****

1. Тестовый пример

Необходимо отсортировать массив [-2, -8, -10, 5, 3] с помощью пирамидальной сортировки.

-2

/ \

-8 -10

/ \

5 3

5 > -8, поэтому меняем эти числа местами

-2

/ \

5 -10

/ \

-8 3

5 > -2, поэтому меняем эти числа местами

5

/ \

-2 -10

/ \

-8 3

5 становится самым большим элементом пирамиды, помещаем его в новый массив, вершиной пирамиды выбираем следующее число -2. Получим массив [5] и новую пирамиду:

-2

/ \

-8 -10

/

3

3 > -8, поэтому меняем эти числа местами

-2

/ \

3 -10

/

-8

3 > -2, поэтому меняем эти числа местами

3

/ \

-2 -10

/

-8

3 становится самым большим элементом пирамиды, помещаем его в массив [5], вершиной пирамиды выбираем следующее число -2. Получим [3, 5]

-2

/ \

-8 -10

-8 < -2 и -10 < -2, оставляем таблицу без изменений, добавляем в массив [3, 5]. Получим массив [-2, 3, 5] и новую пирамиду:

-8

/

-10

-10 > -8, поэтому оставляем пирамиду без изменений, добавляем элемент -8 в массив, получим [-8, -2, 3, 5]. Получаем новую пирамиду:

-10

Добавляем этот элемент, последний в пирамиде, в массив, получим

[-10, -8, -2, 3, 5].

1. Код программы на языке С++

#include <iostream>

#include <ctime>

#include <iomanip>

using namespace std;

int\* Inputarr(int& size)

{

int i, \* arr;

cout << "array size: ";

cin >> size;

arr = new int[size];

for (i = 0; i < size; ++i)

arr[i] = rand() % 99;

return arr;

}

void Showarr(const char\* head, int\* arr, int size)

{

int i;

cout << head;

for (i = 0; i < size; ++i)

cout << setw(3) << arr[i];

cout << endl;

}

void heapify(int arr[], int size, int i)

{

int largest = i;\

int l = 2 \* i + 1; // левый = 2\*i + 1

int r = 2 \* i + 2; // правый = 2\*i + 2

if (l < size && arr[l] > arr[largest])

largest = l;

if (r < size && arr[r] > arr[largest])

largest = r;

if (largest != i)

{

swap(arr[i], arr[largest]);

heapify(arr, size, largest);

}

}

void heapSort(int arr[], int size)

{

for (int i = size / 2 - 1; i >= 0; i--)

heapify(arr, size, i);

for (int i = size - 1; i >= 0; i--)

{

swap(arr[0], arr[i]);

heapify(arr, i, 0);

}

}

int main()

{

int\* arr, size;

clock\_t time1, time2;

srand(time(NULL));

arr = Inputarr(size);

Showarr(" array:\n", arr, size);

time1 = clock();

heapSort(arr, size );

time2 = clock();

Showarr("Sorted array:\n", arr, size);

cout << "time : " << (double)(time2 - time1) / CLOCKS\_PER\_SEC << " seconds\n";

}

1. Результаты программы

5 элементов:

Изображение выглядит как текст

Автоматически созданное описание

100 элементов:

|  |
| --- |
| 1000 элементов: |
|  |
| 10 000 элементов: |
|  |
|  |

1. Вывод

В ходе выполнения лабораторной работы был изучен и реализован алгоритм пирамидальной сортировки массива. Программа работает правильно, работа выполнена в полном объеме.