МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний авіаційний університет

Кафедра прикладної інформатики

Лабараторна робота №8

З дисципліни “Теорія алгоритмів”

Виконав:

Студент 1 курсу ІКІТ

Групи УС-112

Лазоренко Максим Олександрович

Перевірив: Ходаков Д.В.

Київ, 2016

Теоретичні відомості:

Сортировка пузырьком

**Сортировка простыми обменами**, **сортиро́вка пузырько́м** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *bubble sort*) — простой [алгоритм сортировки](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B8). Для понимания и реализации этот алгоритм — простейший, но эффективен он лишь для небольших массивов. Сложность алгоритма: [{\displaystyle O}](https://ru.wikipedia.org/wiki/O_%D0%B1%D0%BE%D0%BB%D1%8C%D1%88%D0%BE%D0%B5){\displaystyle (n^{2})}.

Алгоритм считается учебным и практически не применяется вне учебной литературы, вместо него на практике применяются более эффективные алгоритмы сортировки. В то же время метод сортировки обменами лежит в основе некоторых более совершенных алгоритмов, таких как [шейкерная сортировка](https://ru.wikipedia.org/wiki/%D0%A8%D0%B5%D0%B9%D0%BA%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0" \o "Шейкерная сортировка), [пирамидальная сортировка](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%80%D0%B0%D0%BC%D0%B8%D0%B4%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0) и [быстрая сортировка](https://ru.wikipedia.org/wiki/%D0%91%D1%8B%D1%81%D1%82%D1%80%D0%B0%D1%8F_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0).

Алгоритм

Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются {\displaystyle N-1} раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции, как пузырёк в воде, отсюда и название алгоритма).

## Пример работы алгоритма

Возьмём массив с числами «5 1 4 2 8» и отсортируем значения по возрастанию, используя сортировку пузырьком. Выделены те элементы, которые сравниваются на данном этапе.

Первый проход:

(**5 1** 4 2 8) (**1 5** 4 2 8), Здесь алгоритм сравнивает два первых элемента и меняет их местами.

(1 **5 4** 2 8) (1 **4 5** 2 8), Меняет местами, так как {\displaystyle 5>4}

(1 4 **5 2** 8) (1 4 **2 5** 8), Меняет местами, так как {\displaystyle 5>2}

(1 4 2 **5 8**) (1 4 2 **5 8**), Теперь, ввиду того, что элементы стоят на своих местах ({\displaystyle 8>5}), алгоритм не меняет их местами.

Второй проход:

(**1 4** 2 5 8) (**1 4** 2 5 8)

(1 **4 2** 5 8) (1 **2 4** 5 8), Меняет местами, так как {\displaystyle 4>2}

(1 2 **4 5** 8) (1 2 **4 5** 8)

Теперь массив полностью отсортирован, но алгоритму это неизвестно. Поэтому ему необходимо сделать полный проход и определить, что перестановок элементов не было.

Третий проход:

(**1 2** 4 5 8) (**1 2** 4 5 8)

(1 **2 4** 5 8) (1 **2 4** 5 8)

Теперь массив отсортирован и алгоритм может быть завершён.

**Хід роботи**

**Завдання:** Рассмотрим массив целых чисел а1,...,an. Переставить элементы этого массива так, чтобы после перестановки они были упорядочены по убыванию а1<=а2<=...<=an. отсортировать методом пузырька.

C:\Users\Админ\AppData\Local\Microsoft\Windows\INetCache\Content.Word\nn nnhfg.emf

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int size; // розмір масиву

cout << "Введите количество элементов в масиве" << endl;//ввід кількості елементів масиву

cout << "n = ";

cin >> size;

cout << "Введите поочерёдно элементы массива" << endl;

int \*mas = new int[size]; // виділення пам'яті падмасив

// заповнення масиву

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

cout << "mas[" << i << "] = ";

cin >> mas[i];

}

int temp; // тимчасова змінна для сортування масиву

// Сортування масиву

for (int i = 0; i < size - 1; i++) {

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

if (mas[j] < mas[j + 1]) {

// змінюємо елементи місцями

temp = mas[j];

mas[j] = mas[j + 1];

mas[j + 1] = temp;

}

}

}

// вивід відсотованого масиву на екран

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

cout << mas[i] << " ";

}

cout << endl;

delete[] mas; // звільнення мам'яті;

system("pause");

return 0;

}