Практическое задание №1 - Методы сортировки

1. Постановка задачи

Необходимо упорядочить (отсортировать) по возрастанию (убыванию) массив целых чисел двумя предложенными методами сортировки и сравнить их эффективность по количеству выполненных операций. Методы сортировки определяются вариантом задания (пункт 4). Смысл понятий «перемещение элементов» и «шаг сортировки» зависит от метода сортировки.

В процессе сортировки необходимо оценить вычислительную сложность методов для введенного массива посредством подсчета количества выполненных операций сравнения и количества выполненных операций перемещения элементов массива.

Необходимо предусмотреть возможность демонстрации процесса упорядочения массива в тестовом режиме, т. е. возможность отображения состояния последовательности на разных этапах сортировки.

В программе необходимо предусмотреть «user-friendly» удобный понятный пользовательский интерфейс.

1.1 Входные данные

Программа запрашивает у пользователя:

- Длину упорядочиваемой последовательности ($\leqslant 100$).
- Последовательность целых чисел для сортировки через пробел.
- Тип сортировки (по возрастанию или по убыванию).
- Режим работы программы (обычный или тестовый).

1.2 Выходные данные

В *обычном режиме* программа должна выводить следующие данные:

- Название используемого метода сортировки.
- Исходную последовательность.
- Полученную упорядоченную последовательность.
- Количество выполненных сравнений элементов.
- Количество выполненных перемещений элементов.

В *тестовом режиме* программа должна дополнительно печатать:

- номер шага;
- последовательность после каждого этапа (шага) сортировки;
- сравниваемые элементы;
- текущее количество сравнений;
- текущее количество перемещений.

1.3 Пример работы программы (в обычном режиме)

Пример входных данных:

```
Enter sequence size: 10
Enter sequence: 0 1 2 3 4 5 6 7 8 9
Sort type (1 - increasing sequence, 2 - descending sequence): 1
Program mode (1 – Release, 2 - Debug): 1
```

Ожидаемый результат работы программы:

```
Sort method 1: Bubble sort
Source sequence: 0 1 2 3 4 5 6 7 8 9
Final sequence: 0 1 2 3 4 5 6 7 8 9
Number of comparisons: 0
Number of movements: 0

Sort method 2: Shell sort
Source sequence: 0 1 2 3 4 5 6 7 8 9
Final sequence: 0 1 2 3 4 5 6 7 8 9
Number of comparisons: 0
Number of movements: 0
```

2. Срок выполнения задания:

Последний день сдачи: **16.11.2021**

3. Процедура выполнения и сдачи задания

- Изучить методы сортировки в соответствии с Вашим заданием (разобрать методы на простых примерах)
- Изучить требования к программе и требования к сдаче программы
- Реализовать методы в виде программы
- Протестировать программу
- Сдать программу преподавателю в компьютерном классе
- Отправить программу на почту с темой письма в следующем формате:

Тема письма:

[номер группы] – [Задание №N] – Фамилия Имя

Пример:

[113] – Задание №1 – Иванов Иван

Адрес для отправки работ: practicum.msu@yandex.ru

4. Варианты задания

№	Метод сортировки	Перемещение	Шаг сортировки
1	Сортировка вставками: простые вставки (п. 6.1а)	$d_1 := d_2$	установка на место очередного элемента
2	Сортировка вставками: бинарные вставки (п. 6.1б)	$d_1 := d_2$	установка на место очередного элемента
3	Метод пузырька (п. 6.2)	$d_1 \leftrightarrow d_2$	один просмотр массива с перестановками
4	Челночная сортировка (п. 6.3)	$d_1 \leftrightarrow d_2$	перестановка на окончательное место элемента, нарушающего порядок
5	Сортировка посредством простого выбора (п. 6.4)	$d_1 \leftrightarrow d_2$	установка на место очередного элемента
6	Метод Шелла (п. 6.5)	В зависимости от метода сортировки подмассива	упорядочен подмассив $x_i, x_{i+k}, x_{i+2k}, \ldots$
7	Быстрая сортировка (п. 6.6)	$d_1 \leftrightarrow d_2$	Перестановка элементов подмассива относительно элемента
8	Сортировка слиянием: простое слияние (п. 6.7а)	$d_1 := d_2$	Описан в методе
9	Сортировка слиянием: естественное слияние (п. 6.7б)	$d_1 := d_2$	описан в методе

5. Требования

5.1 Требования к программе

1. Методы сортировки реализовать в виде процедур; массив чисел и его длина — параметры процедуры.
2. Ввод данных и вывод результата оформить в виде соответствующих процедур (функций).
3. Программа должна содержать «user-friendly» удобный понятный пользовательский интерфейс для ввода и вывода данных.
4. Программа должна поддерживать два режима работы (основной и тестовый).
5. Операторы перехода любого вида, включая exit, break, goto не использовать.
6. Текст программы должен быть хорошо читаем, структурирован и оформлен в соответствии с правилами оформления программ на языке Паскаль с отступами (см. приложение к письму).

5.2 Требования к сдаче программы

- Необходимо быть готовым объяснить суть работы реализованного метода сортировки.
- Необходимо иметь в запасе примеры худшего и лучшего массивов (т. е. массивов, на которых достигается максимальное и минимальное количество операций) с подсчётом количества операций для этих массивов (посчитали самостоятельно без компьютера на бумаге), чтобы сравнить с результатом работы программы.

6. Краткое описание методов сортировки

Упорядочивается по возрастанию массив $X_1, \dots, X_n$.

6.1 Сортировки вставками

Общая идея методов сортировки вставками: в ранее упорядоченную подпоследовательность $X_1, X_2, \dots, X_{k-1}$ вставляется элемент X_k так, чтобы упорядоченными оказались k первых элементов исходной последовательности. Для этого:

1. в подмассиве $X_1, X_2, \dots, X_{k-1}$ отыскивается место для элемента X_k, т. е. находится номер i такой, что $X_j < X_k$ для $j \in [1, i]$ и $X_k < X_j$ для $j \in [i+1, k-1]$;
2. элементы $X_{i+1}, \dots, X_{k-1}$ сдвигаются на одну позицию вправо, элемент X_k записывается на место элемента X_{i+1}.

Процесс заканчивается, когда поставлен на своё место элемент X_n.

В зависимости от способа поиска места для элемента X_k различаются следующие методы.

а) ***Метод простых вставок.*** Величина X_k по очереди сравнивается с элементами X_{k-1}, X_{k-2}, X_{k-3} и т. д. до тех пор, пока не будет найдено место для элемента X_k.

б) ***Метод бинарных вставок.*** Величина X_k сравнивается со средним элементом подпоследовательности $X_1, X_2, \dots, X_{k-1}$. Если X_k меньше этого элемента, то место для него ищется тем же способом в левой половине подпоследовательности, а если больше –в правой половине.

6.2 Метод пузырька

По очереди просматриваются пары соседних элементов X_1 и X_2, X_2 и X_3, X_3 и X_4 и т. д.; в неупорядоченных парах ($X_i > X_{i+1}$) переставляются элементы. По окончании просмотра и перестановок максимальный элемент станет последним элементом последовательности. Далее аналогичная процедура применяется ко всем элементам, кроме последнего. Если при очередном просмотре последовательности не было произведено ни одной перестановки элементов, то последовательность уже упорядочена и следует прекратить сортировку.

6.3 Челночная сортировка

В этой сортировке элементы массива просматриваются попеременно то слева направо, то справа налево, как движется челнок в ткацком производстве. Сначала массив просматривается слева направо, сравниваются последовательные пары X_1 и X_2, X_2 и X_3, X_3 и

X_4 и т. д. до обнаружения неупорядоченной пары: $X_{k-1} > X_k$. Далее начинается движение справа налево: «неупорядоченный » элемент X_k сравнивается и меняется местами с предыдущими элементами, пока не окажется на своем месте. Таким образом, получается упорядоченный подмассив $X_1, X_2, \dots, X_k$. Затем продолжается движение слева направо: начиная с $(k + 1)$-ой позиции возобновляется просмотр массива X и поиск неупорядоченной пары. Если неупорядоченная пара не найдется, процесс заканчивается.

6.4Сортировка посредством простого выбора

В массиве $X_1, \dots, X_n$ отыскивается минимальный элемент. Найденный элемент меняется местами с первым элементом массива. Затем так же обрабатывается подмассив $X_2, \dots, X_n$. Когда обработан подмассив X_{n-1}, X_n, сортировка заканчивается.

6.5 Метод Шелла

Пусть k — целое от 1 до $n/2$. Независимо друг от друга упорядочиваются (одним из описанных выше методов) k подпоследовательностей из элементов, отстоящих друг от друга на k позиций: $X_i, X_{i+k}, X_{i+2k}, \dots$ $(i = 1, 2, \dots, k)$.

Затем k уменьшается, и процесс повторяется заново. Последний шаг должен выполняться при $k = 1$.

Значение k можно менять по разным законам. При сдаче программы следует объяснить выбранный способ изменения k. Метод сортировки подмассивов следует описать в виде вложенной процедуры, параметры — индекс начала подмассива и шаг k.

6.6 Быстрая сортировка (метод Хоара)

Суть метода заключается в следующем. Выбирается q — произвольный элемент массива. Элементы массива $X_1, \dots, X_n$ переставляются таким образом, чтобы в начале массива оказались элементы, меньшие или равные q, а в конце — большие или равные q. Таким образом, массив делится на две части.

Требуемую перестановку элементов можно выполнить следующим образом. Последовательность просматривается слева направо, пока не встретится элемент, больший или равный q; затем массив просматривается справа налево до элемента, меньшего или равного q. Найденные элементы меняются местами, после чего просмотры с обоих концов последовательности возобновляются со следующих элементов.

После разбиения массива на две части описанный выше процесс рекурсивно применяется к полученным подмассивам.

Когда длина обрабатываемого подмассива станет меньше трех, его надо упорядочить более простым методом.

6.7 Сортировка слиянием

Основная идея такой сортировки — разделить последовательность на уже упорядоченные подпоследовательности (назовем их «отрезками») и затем объединять эти отрезки во всё более длинные упорядоченные отрезки, пока не получится единая упорядоченная последовательность. Отметим, что при этом необходима дополнительная память (массив $Y[1..n]$).

Различаются следующие варианты сортировки слиянием.

а) ***Простое слияние.*** Считается, что вначале отрезки состоят только из одного элемента, они сливаются в отрезки из двух элементов (из X_1 и X_2, из X_3 и X_4, …), которые переносятся в массив Y. На втором этапе соседние двухэлементные отрезки (Y_1, Y_2 и Y_3, Y_4; Y_5, Y_6 и Y_7, Y_8; …) объединяются в отрезки из 4 элементов, которые записываются в массив X. На третьем этапе строятся отрезки из 8 элементов, которые заносятся в массив Y, и т. д.

б) ***Естественное слияние.*** Массив X просматривается с начала, определяется наиболее длинный упорядоченный по неубыванию отрезок; затем массив X просматривается с конца и выбирается наиболее длинный упорядоченный (также по неубыванию), отрезок в конце массива; полученные отрезки сливаются в один отрезок, который записывается в начало массива Y. Затем сливаются следующие максимально длинные отрезки с обоих концов, и полученный отрезок записывается (справа налево) в конец массива Y. Третьи по порядку отрезки после слияния записываются снова в начало Y (вслед за первым объединенным отрезком), четвертые — в конец Y (перед вторым объединенным отрезком) и т. д. Первый этап сортировки оканчивается, когда все элементы из X будут перенесены в массив Y. На втором этапе применяется та же процедура, только массивы X и Y меняются ролями. И так далее, пока весь массив не образует один упорядоченный отрезок.

Замечание: в обоих вариантах следует учитывать, что в конце концов упорядоченные элементы должны оказаться в массиве X.