

Лабораторная работа № 2

Тема: «Алгоритмы сортировки»

Цель работы: Изучение алгоритмов сортировки

Сортировка вставками

Работает, проходя по массиву и перемещая нужное значение в начало массива. После того, как обработана очередная позиция, мы знаем, что все позиции до нее отсортированы, а после нее — нет. Сортировка вставками обрабатывает элементы массива по порядку.

Принцип данной сортировки заключается в том, что берется число перемещаясь вправо, сверяя значения. Числа больше нужного - передвигаются вправо, числа меньше - остаются в левой части и проверяя условия больше либо меньше конкретного числа.

Сортировка вставками наиболее эффективна когда массив уже частично отсортирован и когда элементов массива не много. Если же элементов меньше 10 то данный алгоритм является лучшим.

Сортировка пузырьком

Это самый простой алгоритм сортировки. Он проходит по массиву несколько раз, на каждом этапе перемещая самое большое значение из неотсортированных в конец массива.

При первом проходе по массиву мы сравниваем необходимые значения. Поскольку нужное число больше текущего, мы оставляем их как есть. После чего сравниваем большее число и меньшее. Меньше число меньше, поэтому мы меняем их местами, перемещая число на одну позицию ближе к концу массива.

Если предположить, что в массиве содержится N элементов и хотя бы один из них занимает свое место в результате однократного пересмотра значений, то алгоритм может совершить не более N проходов. (Все N понадобятся, когда массив изначально отсортирован в обратном порядке.) Каждое такое прохождение включает N шагов, отсюда общее время работы алгоритма — $O(N^2)$.

Пузырьковая сортировка часто рассматривается как наиболее неэффективный сортировочный метод, поскольку она должна переставлять элементы до того, как станет известна их окончательная позиция.

					АиСД 090000.000 ПР								
Изм	Лист	№ докум.	Подпись	Дата									
Разраб.		Климова Ю.В.			Лабораторная работа № 2 «Алгоритмы сортировки»				Лит	Лист	Листов		
Провер.		Береза А.Н.									1	8	
									ИСОиП(ф)ДГТУ ИСТ-Тб21				
Н.контр.													
Утв.													



Рисунок 1-Блок-схема алгоритма Вставки

Программный код алгоритма Вставки

```
#include <stdio.h>

void selectionSort(int *num, int size)
{
    int min, temp;
    for (int i = 0; i < size - 1; i++)
    {
        min = i;
        for (int j = i + 1; j < size; j++)
        {
            if (num[j] < num[min])
                min = j;
        }
        temp = num[i];
        num[i] = num[min];
        num[min] = temp;
    }
}

int main()
{
    int a[10];

    for (int i = 0; i < 10; i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    selectionSort(a, 10);  for (int i = 0; i < 10; i++)
        printf("%d ", a[i]);
    getchar(); getchar();
    return 0;
}
```

Результат выполнения алгоритма:

```
a[0] = 5
a[1] = 6
a[2] = 24
a[3] = 74
a[4] = 53
a[5] = 9
a[6] = 462
a[7] = 0
a[8] = 47
a[9] = 36
0 5 6 9 24 36 47 53 74 462 □
```

					100000.000 ПР	Лист
						3
Изм	Лист	№ докум.	Подпись	Дата		

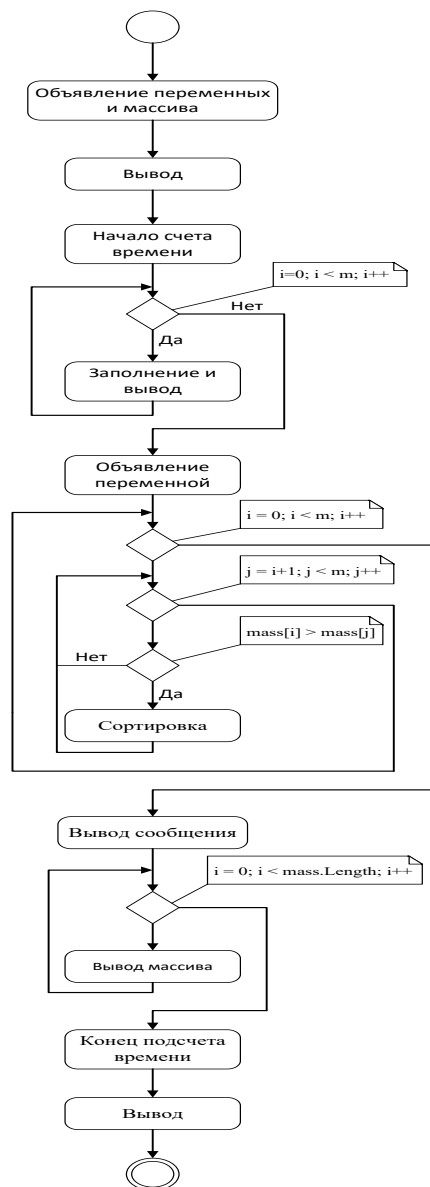


Рисунок 2-Блок-схема алгоритма сортировки Пузырек

Программный код алгоритма Пузырек

					100000.000 ПР	Лист
Изм	Лист	№ докум.	Подпись	Дата		4

```

#include <stdio.h>

// Функция сортировки прямым обменом (метод "пузырька")
void bubbleSort(int *num, int size)
{
    // Для всех элементов
    for (int i = 0; i < size - 1; i++)
    {
        for (int j = (size - 1); j > i; j--) // для всех элементов после i-ого
        {
            if (num[j - 1] > num[j]) // если текущий элемент меньше предыдущего
            {
                int temp = num[j - 1]; // меняем их местами
                num[j - 1] = num[j];
                num[j] = temp;
            }
        }
    }
}

int main()
{
    int a[10]; // Объявляем массив из 10 элементов
    // Вводим значения элементов массива
    for (int i = 0; i < 10; i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    bubbleSort(a, 10); // вызываем функцию сортировки
    // Выводим отсортированные элементы массива
    for (int i = 0; i < 10; i++)
        printf("%d ", a[i]);
    getchar(); getchar();
    return 0;
}

```

Результат выполнения алгоритм:

					100000.000 ПР	Лист
						5
Изм	Лист	№ докум.	Подпись	Дата		

```
a[0] = 66
a[1] = 95
a[2] = 52
a[3] = 69
a[4] = 2
a[5] = 58
a[6] = 47
a[7] = 35
a[8] = 365
a[9] = 36
2 35 36 47 52 58 66 69 95 365
```