

Практическая работа № 2

Алгоритмы сортировки: Сортировка вставками

Цель работы: Изучить способы применения Алгоритмов сортировки: Сортировки вставками

Оборудование: Windows 10, Visual Studio, Microsoft Word.

КРАТКАЯ ТЕОРИЯ

Сортировка вставками – простой алгоритм сортировки, преимущественно использующийся в учебном программировании. К положительной стороне метода относится простота реализации, а также его эффективность на частично упорядоченных последовательностях, и/или состоящих из небольшого числа элементов. Тем не менее, высокая вычислительная сложность не позволяет рекомендовать алгоритм в повсеместном использовании.

Рассмотрим алгоритм сортировки вставками на примере колоды игровых карт. Процесс их упорядочивания по возрастанию (в колоде карты расположены в случайном порядке) будет следующим. Обратим внимание на вторую карту, если ее значение меньше первой, то меняем эти карты местами, в противном случае карты сохраняют свои позиции, и алгоритм переходит к шагу 2. На 2-ом шаге смотрим на третью карту, здесь возможны четыре случая отношения значений карт:

1. первая и вторая карта меньше третьей;
2. первая и вторая карта больше третьей;
3. первая карта уступает значением третьей, а вторая превосходит ее;
4. первая карта превосходит значением третью карту, а вторая уступает ей.

АИСД.09.03.02.110000.00001 IP							
Изм	Лист	№ докум.	Подпись	Дата			
Разраб.		Куличенко Е.В.			Практическая работа №2 «Алгоритмы сортировки: Сортировка вставками	Лит	Лист
Провер.		Береза А.Н.					Листов
							1
Н.контр.						ИСОuП(ф) ДГТУ ИСТ-Tb21	
Утв.							

В первом случае не происходит никаких перестановок. Во втором – вторая карта смещается на место третьей, первая на место второй, а третья карта занимает позицию первой.

В предпоследнем случае первая карта остается на своем месте, в то время как вторая и третья меняются местами. Ну и наконец, последний случай требует рокировки лишь первой и третьей карт. Все последующие шаги полностью аналогичны расписанным выше.

Время выполнения алгоритма зависит от входных данных: чем большее множество нужно отсортировать, тем большее время выполняется сортировка. Также на время выполнения влияет исходная упорядоченность массива. Так, лучшим случаем является отсортированный массив, а худшим — массив, отсортированный в порядке, обратном нужному. Временная сложность алгоритма при худшем варианте входных данных — $O(n^2)$.

ИСХОДНЫЙ КОД

```
#include <iostream>
using namespace std;

int main()
{
    const int N = 10;
    int a[N] = { 12, 5, 3, 2, 45, 96, 6, 8, 11, 24 };

    int buff = 0; // для хранения перемещаемого значения
    int i, j;      // для циклов

    /***** Начало сортировки *****/
    for (i = 1; i < N; i++)
    {
        buff = a[i]; // запомним обрабатываемый элемент
        // и начнем перемещение элементов слева от него
        // пока запомненный не окажется меньше чем перемещаемый
        for (j = i - 1; j >= 0 && a[j] > buff; j--)
            a[j + 1] = a[j];

        a[j + 1] = buff; // и поставим запомненный на его новое место
    }
    /***** Конец сортировки *****/

    for (int i = 0; i < N; i++) // вывод отсортированного массива
        cout << a[i] << '\t';
    cout << endl;
}
```

```
Консоль отладки Microsoft Visual Studio

2      3      5      6      8      11     12     24     45     96

C:\Users\Lisa Cool\source\repos\ConsoleApplication8\Debug\ConsoleApplication8.exe (
м 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сер
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

					ИКСС.09.03.02.110000.0000ПР	Лист
						3
Изм	Лист	№ докум.	Подпись	Дата		