

ИССЛЕДОВАНИЕ И ОЦЕНКА АЛГОРИТМОВ СОРТИРОВКИ

Цель работы: Разработка программ, реализующих различные алгоритмы сортировки, и оценка их временной и пространственной сложности.

Словесное описание алгоритма: в лабораторной работе реализовано 2 алгоритма сортировки: ускоренный алгоритм сортировки пузырьком и сортировка вставками. У алгоритм сортировки пузырьком сложность выполнения всегда $O(n^2)$. Этот алгоритм можно оптимизировать остановив внутренний цикл, если там не ожидается больше перестановок. Сортировка вставками - это простой алгоритм сортировки, который работает так же, как мы сортируем игральные карты в руках ($O(n^2)$).

Текст программы:

```
public static void bubbleSortOptimized(int[] notSortedList){
    boolean swapped;
    int n = notSortedList.length;
    for(int i = 0; i < n; i++){
        swapped = false;
        for(int j = 0; j < n-i-1; j++){
            if(notSortedList[j] > notSortedList[j+1]){
                int tmp = notSortedList[j];
                notSortedList[j] = notSortedList[j+1];
                notSortedList[j+1] = tmp;
                swapped = true;
            }
        }
        if(swapped == false)
            break;
    }
}
```

```
public static void insertionSort(int[] list){
    int n = list.length;
    for (int i = 0; i < n; i++) {
        int key = list[i];
        int j = i-1;

        while(j >= 0 && list[j] > key){
            list[j+1] = list[j];
            j--;
        }

        list[j+1] = key;
    }
}
```

Формулы верхней оценки временной и ёмкостной сложности:

Ускоренный алгоритм сортировки: $13n^2 + 4n + 1$ (Худший случай тогда, когда минимальный элемент находится в конце)

Сортировка вставками: $8n^2 + 12n + 1$ (Худший случай тогда, когда минимальный элемент находится в конце)

Результаты экспериментальной оценки временной и емкостной сложности заданных алгоритмов.

