**Домашняя работа сортировка массивов Sort**

Для всех:

Создать класс Sort с реализацией метода bubbleSort (если сложно, посмотрите справку)

public class Sort {  
 public static void bubbleSort(int[] arrayNums) {  
  
 }  
}

В классе Main создать массив и проверить как он работает.

Дополнительные задания:

1 - Вставками (Сырков Илья) insertionSort

2 - Быстрая сортировка (Вожегов Андрей) quickSort

3 - Слиянием (Тарасов Владислав) mergeSort

4 - Выбором (Баранов Валерий) selectionSort

5 - Диаграмма времени исполнения методов сортировок (Коптева Елена)

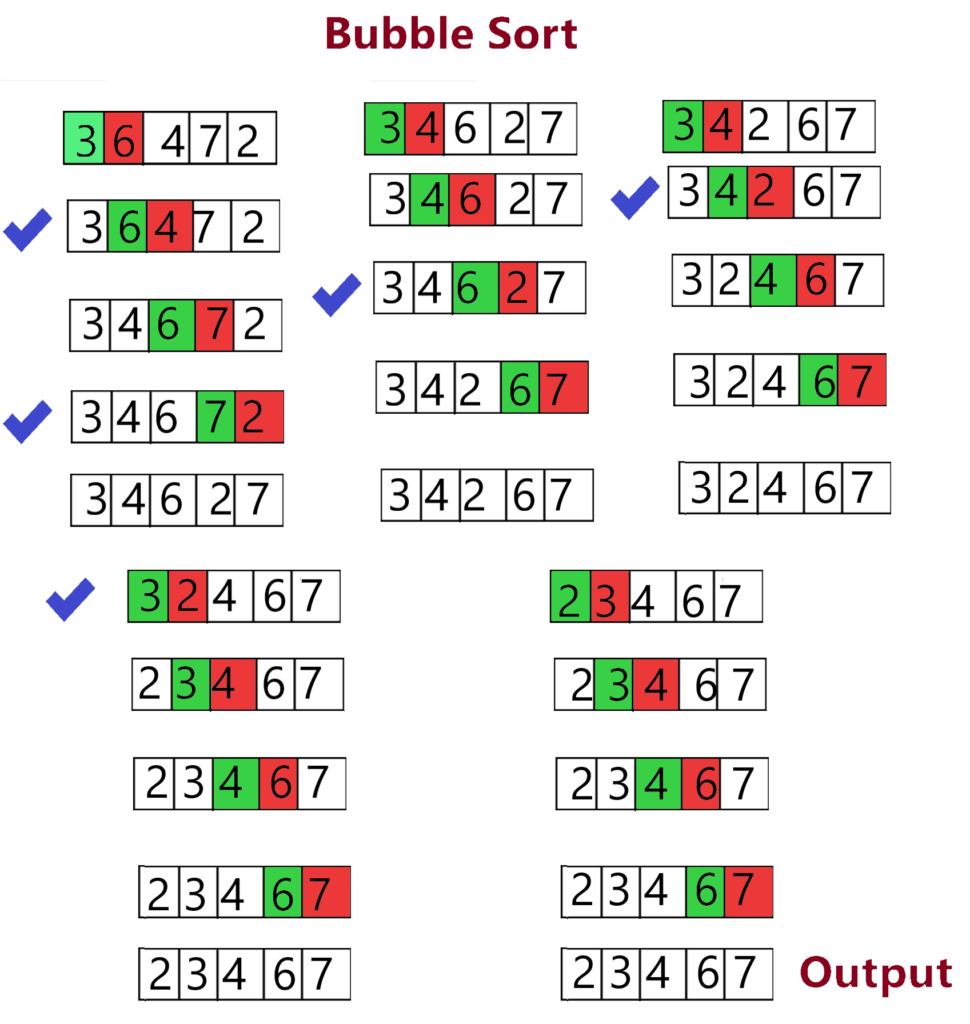
(1-4) Необходимо показать наглядно алгоритм сортировки, рассказать, как он работает, сказать/показать сложность алгоритма в лучшем, среднем и худшем случае. Написать функцию данного алгоритма и быть готовым объяснить каждую строчку.

(5) Сравнить сортировку пузырьком и любую из (1-4) сортировок на скорость, со 2-го запуска, т.к. в первый может программа долго компилироваться. Есть справка для отчета времени.

(1-5) Показывать будете на своем мониторе. Удачи!

**Справка:**

В пузырьковой сортировке1-ый элемент сравнивается со 2-ым и меньший из них становится на место 1-го, а больший на место 2-го и так далее до конца, после прохождения первого цикла, он проходит его еще и еще определенное количество раз.



(5) В Java есть несколько способов измерения времени выполнения определенного участка кода. Один из способов - использовать класс System с его методами currentTimeMillis() или nanoTime(). Например, вы можете начать отсчет времени до выполнения функции, затем вызвать эту функцию и, наконец, закончить отсчет времени.

Вот пример:

public static void main(String[] args) {

long startTime = System.currentTimeMillis(); // Или System.nanoTime() для более точного времени

// Вызываете функцию, для которой хотите измерить время выполнения

yourFunction();

long endTime = System.currentTimeMillis(); // Или System.nanoTime() для более точного времени

long executionTime = endTime - startTime;

System.out.println("Время выполнения функции: " + executionTime + " миллисекунд"); // Или вывод в наносекундах для nanoTime()

}

// Пример вашей функции

public static void yourFunction() {

// Ваш код

}

Этот код измеряет время выполнения yourFunction() путем получения времени до и после вызова функции и нахождения разницы между ними. Важно помнить, что измерение времени с помощью nanoTime() может быть более точным, но обычно currentTimeMillis() хорошо подходит для большинства случаев.