Реализовать многопоточную версию Пузырьковой сортировки. Обеспечить необходимую синхронизацию потоков. Должна быть возможность ограничить максимальное кол-во потоков, которые одновременно могут быть запущены. Для ограничения числа потоков воспользоваться классом Semaphore. Для повторного использования потоков воспользоваться классом ThreadPool (если в метод сортировки экземпляр ThreadPool передается, то пользоваться им, иначе создавать нити самостоятельно).

Для демонстрации работы метода писать (например в консоль) информативные логи о последовательности отработки заданий в потоках.

Сравнить эффективность многопоточной версии сортировки (естественно при сравнении никакие логи не писать) с однопоточной (реализовать отдельно, без обращения к потокам вообще) с различными параметрами, сделать выводы.

При желании можно воспользоваться классами из пространства имен System.Threading.Tasks.

Комментарий:

Потоки могут поднимать (опускать) наименьший (наибольший) элемент параллельно. Синхронизация необходима, чтобы гарантировать, что M+1 поток, будет сравнивать и менять элементы [N] и [N+1] только после того, как поток M уже сравнил и поменял элементы [N+1] и [N+2].