Пояснения.

Очевидно, что существует проблема, связанная с примением пула потоков при использовании рекурсивных алгоритмов. Очевидно также, что для некоторых рекурсивных алгоритмов эта проблема решается просто.

Алгоритм быстрой сортировки подготавливает для дальнейшей сортировки два раздела и далее рекурсивно сортирует разделы. Если выполнение очередного рекурсивного вызова выполняется задачей в пуле, то ясно, что во избежание блокировки на уровне рекурсии n после подготовки разделов необходимо дать задания на сортировку разделов и завершить выполнение функции на уровне n, чтобы не занимать поток в пуле. Проблема заключается в том, что основной поток должен дождаться завершения выполнения всех заданий.

Достичь этого можно самыми разными способами. Я предлагаю решение не простое, а очень простое. Метод sort класса QuickSorter применяется для сортировки больших массивов, он создает локальный пул и вызывает метод quicksort, который при большом размере разделов помещает задания на сортировку разделов в пул. Деструктор класса ThreadPool задерживает основной поток до завершения выполнения всех заданий. Очевидно, что до выполнения всех заданий число заданий в пуле не будет нулевым, т.к. на каждом уровне рекурсии quiсksort, выполняющийся в пуле, увеличивает число заданий. Метод num\_Task позволяет убедиться в том, что 4 потока пула выполнили более десятка заданий, проверка показывает, что исходный массив отсортирован.