

## Задание:

5. Ответьте на вопросы: почему при небольших значениях `sz` цикл `cilk_for` уступает циклу `for` в быстродействии? В каких случаях целесообразно использовать цикл `cilk_for`? В чем принципиальное отличие параллелизации с использованием `cilk_for` от параллелизации с использованием `cilk_spawn` в паре с `cilk_sync`?

## Ответ:

1. Это происходит, потому что накладные расходы на реализацию многопоточности превышают выигрыш от ее использования
  2. Это происходит, если количество вычислений достаточно велико - выигрыш от использования многопоточности превышает накладные расходы
  3. `cilk_for` - в процессе компиляции тело цикла конвертируется в функцию, которая вызывается рекурсивно в соответствии со стратегией «разделяй и властвуй». Планировщик автоматически распределяет поддеревья рекурсии между обработчиками
- `cilk_spawn` - ручное управление параллелизацией вычислений. Нагрузка между обработчиками не сбалансирована, т.к. каждый дочерний обработчик выполняет только одну итерацию