Name: _____

IB Computer Science 2
Parallelism worksheet

Whereas concurrency is executing tasks during overlapping time windows, **parallelism** is executing them truly simultaneously. Parallelism therefore requires multiple CPU cores, whether on the same processor, multiple processors on the same motherboard, or multiple servers connected via a network to form a supercomputer or distributed system.

Download our merge sort implementation from the recursion unit, including the merge algorithm that it depends on. Place them in the folder where you usually write activity code.

1. Remove all the prints and time the `mergeSort()` method call using `System.currentTimeMillis()`. Run it to ensure that the only output is the measurement. How long does it take to sort the array?

2. Increase the array size to 100 million elements, or as large as you can make it without your machine barfing. How long does it take to sort this array?

Make a copy of the program called `ParallelMergeSort.java`. Create a global thread pool like we just did as a class. Modify your `mergeSort()` method implementation to `.submit()` `Callable<>`s to the global thread pool instead of directly recursing, then `.get()` their results just before performing the merge step.

3. Once you have resolved the compiler warnings, run the program. What happens? Looking at the warning messages, what do you think is going wrong?

Kill the program by hitting Ctrl+Z then running the command `kill -kill %1`. We will need to be more careful about the number of threads we spawn, to avoid overwhelming the operating system with an exponential explosion in the number of running tasks…

4. Replace the `Executors.newCachedThreadPool()` call with an `Executors.newWorkStealingPool()`, which caps the number of threads at the number of CPU cores (the maximum possible parallelism). How long does it take now? *Note: If it is not faster, discuss your code as a group to ensure it makes sense. If you are stuck, seek assistance from a group that has finished.*

5. What features of the merge sort algorithm make it well suited to parallelization?