Environment:

All executions tests were performed in the same environment, a computer equipped with an 8 core (all locked at 4.2 GHz), 16 threads AMD Ryzen 7 3700X. I used the latest version of Eclipse and the background processes were kept to a minimum.

Knapsack:

For the sorting algorithm I decided to use a merge sort algorithm. Being a recursive algorithm I decided to parallelize it by implementing a Fork Join parallelization. The task will keep being divided in two till the size of the array meets a certain value, in this tests the value used was 8192. Regarding granularity I tried to use the number of queued surplus tasks to determine when the task should be done sequentially but the returned value never goes over 8 and making the task divide when the queued surplus task count was above 7 it worsen the execution time. The returned values of the number of queued tasks, in my tests, never returned a value above 2. With that being said, granularity was not used so that the if statement in line 24 of the class ParallelMergeSort would have less variables to evaluate.