Jack Diaz
111499298
Design Document
Project 3

The way I designed my algorithm was heavily based off of the single threaded algorithm provided. I submitted a task to an executor for each word in the dictionary. This executor would spawn a number of threads equal to the number of available processors plus one. Each thread would find instances of its word in the puzzle and add them to the paths ConcurrentLinkedQueue. This structure is thread safe so I don't have to worry if two threads try to add words to this Queue at the same time.

Once all the threads terminated, or a specified ratio of the total given time is up, then we start making a solution. This part of the work is serial, but my algorithm seems to do much better than the Single Threaded so I'm not too worried. First we sort all of the paths found by their scores according to the dictionary. This is a greedy approach. We then add the words in score order (highest first) to the solution, and add to filled any points that these words take up so that future words that use those points cannot be added. I was trying to go for a fairly simple approach here as it would be easiest to program and debug.