

## Lab 1: Juice Bottler

### Explanation of the Program:

When running, the ANT build, you are running the main method in Plant. First, the plant initiates itself and a second plant friend, each running on their own thread. Each plant then begins work, and a team of four workers is set up on their own threads. Each worker is specialized at their job, one for fetching oranges, one for peeling, one to squeeze, and one to bottle the juice. The oranges, generated and given an initial state of "Fetched", move through each worker, changing states along the way. To keep it all organized, each plant uses its own queues to keep track of the oranges in each state. The program, and by extension the workers, run for a set amount of time similar to a real work day. When it's time to clock out, the workers finish up whatever was in their queue, then join their threads together. From there, the Plant prints out the summary of the day's events before shutting themselves down.

### Meeting the Requirements

This meets the requirements for the assignment in two ways. I achieve task parallelization through each of the workers working on their own threads all making progress together. When it's time to shut down, they all join and continue working, helping each other, until the last one is done. From there, they join to the main Plant thread, which shuts itself down. Speaking of Plants, I achieve data-parallelization through the use of two plants making independent progress on their own oranges, with their own workers, simultaneously. They keep track of their own workers and orange juicing progress through the use of Concurrent Linked Lists, which are intended to be thread safe. Each plant makes progress on its own work, remaining independent of one another.

### Challenges

Originally, the biggest challenge was figuring out what exactly was needed from me in regards to having this lab completed. The preshow of the mutex did help point in the right direction, but I really needed a visual example so I searched online and found previous students' GitHub's and began running theirs to really understand what they were doing and how it was doing it. I did take inspiration from each of these repos (listed in README under Sources), but, hilariously, given that *some* of those repos didn't even work, I refrained from copying from them. The next biggest challenge was getting it to not only run, but printout correctly, which is why I have so many println's in my final code. Catching the queues backing up was a pretty triumphant milestone, and figuring out the final calculation did make me happy. All in all, I don't think this project was terribly hard, but a visual example of what we need to achieve may help us better understand the assignment.