Elijah Smith
COP4520 HW3
P1 – Birthday Presents Party

For the servants to have more presents than notes, it is likely that servants were adding/removing presents that involved a linkage to or from the same present. Because they were both operating at the same time, one or more presents became improperly linked to the chain and were not accessible from the head. As a result, when the head finally ran out of properly linked presents, the improperly linked presents remained without notes written for them.

In this implementation of a concurrent linked list, we use a lazy linked list to efficiently prevent any hiccups in the adding, removing, or searching process. When a worker goes to remove a present, he claims the lock on the present and the previous present, first marks it so that other servants can continue searching for presents of interest, then finally physically unlinks the present so that they can write a thank you note. No presents are ever improperly linked and are always accessible from the head of the chain, so the invariant holds. The program also validates at the end of execution that a total of 500,000 presents have been both added and removed from the chain between the four workers and that each present is only added and removed once.

Below is the output of executing this program. There are no parameters that are tweakable for experimentation based on the problem description, but the average execution time between runs is between 300ms-500ms.