

[Home/Announcements](#)  
[Style Guide](#)  
[Materials from Class](#)  
[Assignments](#)  
[Schedule](#)  
[Other Resources](#)  
[Submission](#)  
[Syllabus](#)



## Concurrency 3: Categorical Mutual Exclusion

**Due: End of recitation, Week 7**

The purpose of these concurrent programming exercises is to hone your skills in thinking in parallel. This is a very important skill, to the point where many companies view it as being as fundamental as basic algebra or being able to write code. You will be writing a solution to the problem in class, without the use of computers. This will be submitted prior to leaving the class today. Make sure you keep a copy though, as you will implement your solution over the course of the next week.

Three kinds of threads share access to a singly-linked list: searchers, inserters and deleters. Searchers merely examine the list; hence they can execute concurrently with each other. Inserters add new items to the end of the list; insertions must be mutually exclusive to preclude two inserters from inserting new items at about the same time. However, one insert can proceed in parallel with any number of searches. Finally, deleters remove items from anywhere in the list. At most one deleter process can access the list at a time, and deletion must also be mutually exclusive with searches and insertions.

Write concurrent code in any language you like for searchers, inserters and deleters that enforces this kind of three-way categorical mutual exclusion. Remember, some languages have the constructs for concurrency but do NOT actually execute in parallel. Be aware of this and choose wisely.

---

***D. Kevin McGrath***

*Last modified: Tue May 9 07:56:03 2017*