

## **COP 4520**

**Spring 2019**

### **Algorithm Assignments**

Please find the list of algorithm assignments.

Next to every team name, you will find a link to the primary research paper that you are supposed to study and re-implement its algorithms.

Team 1:

“Scalable Lock-Free Vector with Combining”

URL: <https://ieeexplore.ieee.org/document/7967182/>

Team 2:

“An Unbounded Nonblocking Double-Ended Queue”

URL: <https://ieeexplore.ieee.org/document/7573821/>

Team 3:

“Dynamic-sized nonblocking hash tables”

URL: <https://dl.acm.org/citation.cfm?id=2611495>

Note: implement lock-free version

Team 4:

“An Efficient Wait-free Resizable Hash Table”

URL: <https://dl.acm.org/citation.cfm?id=3210408>

Team 5:

“Cache-tries: concurrent lock-free hash tries with constant-time operations”

URL: <https://dl.acm.org/citation.cfm?id=3178498>

Team 6:

“Lock-free Contention Adapting Search Trees”

URL: <http://user.it.uu.se/~bengt/Papers/Full/spaa18.pdf>

Team 7:

“Practical Concurrent Traversals in Search Trees”

URL: <https://www.sri.inf.ethz.ch/papers/ppopp18.pdf>

Team 8:

“Concurrent Linearizable Nearest Neighbour Search in LockFree-kD-tree”

URL: <https://dl.acm.org/citation.cfm?id=3154307>

Team 9:

“Practical Non-blocking Unordered Lists”

URL: <https://dl.acm.org/citation.cfm?id=2950142>

Team 10:

“The lock-free k-LSM relaxed priority queue Lists”

URL: <https://dl.acm.org/citation.cfm?id=2688547>

Team 11:

“Lock-Free Cuckoo Hashing”

URL: <https://ieeexplore.ieee.org/document/6888938/>

Team 12:

“Efficient lock-free binary search trees”

URL: <https://dl.acm.org/citation.cfm?id=2611500>

Team 13:

“Fast and Scalable, Lock-free k-FIFO Queues”

[http://link.springer.com/chapter/10.1007%2F978-3-642-39958-9\\_18](http://link.springer.com/chapter/10.1007%2F978-3-642-39958-9_18)

Team 14:

“A lock-free B+tree”

URL: <http://dl.acm.org/citation.cfm?id=2312016>

Team 15:

“Wait-free linked-lists”

URL: <http://dl.acm.org/citation.cfm?doid=2145816.2145869>

Team 16:

“A lock-free, array-based priority queue”

URL: <http://dl.acm.org/citation.cfm?id=2145876>

Team 17:

“BQ: A Lock-Free Queue with Batching”

<https://dl.acm.org/citation.cfm?id=3210388>

Team 18:

“The Adaptive Priority Queue with Elimination and Combining”

URL: [http://link.springer.com/chapter/10.1007/978-3-662-45174-8\\_28](http://link.springer.com/chapter/10.1007/978-3-662-45174-8_28)

Team 19:

“A Fast Lock-Free Internal Binary Search Tree”

URL: <http://dl.acm.org/citation.cfm?id=2684472>

Team 20:

“A Scalable, Correct Time-Stamped Stack”

<https://dl.acm.org/citation.cfm?id=2676963>

Team 21:

“Concurrent Wait-Free Red Black Trees”

URL: [http://link.springer.com/chapter/10.1007/978-3-319-03089-0\\_4](http://link.springer.com/chapter/10.1007/978-3-319-03089-0_4)

Team 22:

“FA-Stack: A Fast Array-Based Stack with Wait-Free Progress Guarantee”

<https://www.computer.org/csdl/trans/td/2018/04/08097018-abs.html>

Team 23:

“Lock-Free Resizeable Concurrent Tries”

URL: [http://link.springer.com/chapter/10.1007/978-3-642-36036-7\\_11](http://link.springer.com/chapter/10.1007/978-3-642-36036-7_11)

Note: You may request to replace your assigned data structure to one of the extras listed below at any time before Friday, February 15<sup>th</sup> 2019 by 11:59pm. Please send your request via e-mail to Dr. Dechev. Your request will be considered if sent before the deadline but there is no guarantee that it will be granted.

Extra 1:

“Efficient Lock-free Binary Search Trees”

<https://dl.acm.org/citation.cfm?id=2611500>

Extra 2:

“Improving efficacy of concurrent internal binary search trees using local recovery”

<https://dl.acm.org/citation.cfm?id=3288615>

Extra 3:

“Non-Blocking Doubly-Linked Lists with Good Amortized Complexity”

<http://arxiv.org/abs/1408.1935>

Extra 4:

“The Baskets Queue”

URL: [https://link.springer.com/chapter/10.1007/978-3-540-77096-1\\_29](https://link.springer.com/chapter/10.1007/978-3-540-77096-1_29)