

The topic of the next three lectures is cache-efficient data structures. A classic result here is that B-trees are good at exploiting data that is transferred in blocks between cache and main memory, and between main memory and disk, and so on. B-trees achieve $O(\log_B N)$ insert/delete/predecessor/successor for N items and memory block transfers of size B . What's more recent and surprising is that you can achieve the same performance even if you don't know what B is, or in other words, simultaneously for all architectures with all values of B . This is a result in the "cache-oblivious" model, and we'll see how to do it over the next two lectures. Cache-oblivious data structures show some exciting practical promise, too, as they form the foundation of a database startup, [Tokutek](#).

MIT OpenCourseWare
<http://ocw.mit.edu>

6.851 Advanced Data Structures
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.