Cuckoo Hashing with perfect rehash

Vinícius Gusmão Pereira de Sá; Judismar Arpini Junior

December 13, 2022 (Tuesday), 12:30 - Room 1303

The time to insert a key in the classic cuckoo hashing scheme is a random variable that may assume arbitrarily big values, owing to the strictly positive probability that an endless sequence of rehashes take place --- the worst-case is infinite. We propose a cuckoo hashing variant in which the worst-case insertion time is polynomial. To accomplish this, we use two basic ideas. The first is to employ a perfect hashing method on one of the tables whenever a rehash is called for (a perfect rehash, so to speak). The second idea is to make it so that the number of underlying hash tables is no longer constant, but rather an appropriate function of the number of keys. The price to pay is a larger lookup time, which is no longer constant, but doubly logarithmic. Preventing infinite worst-case times is not new in the literature, e.g. a Las Vegas algorithm can be converted into a Monte Carlo algorithm to yield finite predictable time, for the price of some positive, albeit controllable, error probability. Our insertion algorithm follows a random walk approach. When a predefined limit of iterations is reached, we pick a table with some empty slot and rehash all its entries in perfect fashion, including the new key being inserted. With the perfect rehash approach, all cuckoo hashing operations become predictable and finite. Our variant is not dominated by any existing data structure we know of (e.g., the classical cuckoo hashing, the standard hashing with collision handling via synonym lists, or even, say, self-balancing binary search trees). While it cannot be claimed to be an undisputed improvement over those traditional data structures (and it surely has disadvantages as well as advantages when compared to each one of them), it may suit well applications where a highly competitive average-case performance for lookup and insertion operations is desired, without prescinding from a predictable, polynomial bound for their worst-case behavior.

Keywords: Hashing; Data structures; Computational complexity; Algorithms