

Textbooks commonly only mention the relationships between pairs, for example selection sort is equivalent to the worst case of quicksort. One can study sorting algorithms in a more unified, systematic manner. This is done with deriving from PQ sort.

The main tool the paper uses for this is “ d -heap”. “A d -heap is a heap priority queue with a branch factor of d ”.

“ d ” can be seen as a parameter that adjust the messiness of a heap. The paper expresses this as “A d -heap maintains a partial order of the items in the form of rooted tree. For $d = 1$, the order is total (a degenerate tree of sorted values with no branching), and for $d = n-1$, it’s not much of an order at all!”

Black box is used for an analogy for priority queue. Adopt d -heaps as implementation tools and you have:

1. insertion sort for $d = 1$
2. heap sort for $d = 2$
3. selection sort for $d = n - 1$

With observing an interrelation between the two abstract data structures binary search tree and priority queues, one can discover a new sorting algorithm: tree sort.

[...]

Insertion sort is just a merge sort where one splits the n items into $n-1$ and 1 items. Or equivalently, merge sort is just an improvement of insertion sort.