

## Post-midterm homework data structure runtimes

Below are the tight Oh bound runtimes for the best cases and worst cases for the hw data structures after the midterm. The tight bound for average case is not well defined, but the best and worse case bounds are tight. Note that average case analysis is something particularly tricky that we haven't delved into much this course. The worst case and best case runtimes are usually what we would expect you to be able to reason about.

Data Structure and method	Best Case	Average Case	Worst Case
ArrayHeap - insert	$O(1)$ - ex: if you insert a new biggest value in a min heap, no percolateUps happen	$O(\log(n))$	$O(n)$ - resizing
ArrayHeap - removeMin	$O(1)$ - ex: if all values in the heap are the same, so no percolateDown after replacing the root	$O(\log(n))$	$O(\log(n))$
ArrayHeap - peekMin	$O(1)$	$O(1)$	$O(1)$
ArrayDisjointSet - makeSet	$O(1)$	$O(1)$	$O(n)$ - resizing
ArrayDisjointSet - findSet	$O(1)$ - roughly constant height because of optimizations	almost/essentially $O(1)$	$O(\log(n))$ - union by rank optimization guarantees $\log(n)$ height
ArrayDisjointSet - union	$O(1)$ - all the computation is in getting the representatives (calling findSet)	almost/essentially $O(1)$	$O(\log(n))$