

Data Structure Costs

Computing

What each structure is fast at, and what it is slow at — how to choose.

Average-case cost. Every structure trades one operation against another; none wins at everything.

Structure	Access	Search / Insert
Array	$O(1)$	$O(n)$ / $O(n)$
Dynamic array	$O(1)$	$O(n)$ / $O(1)^*$
Linked list	$O(n)$	$O(n)$ / $O(1)$
Hash table	—	$O(1)$ / $O(1)$
Binary search tree	$O(\log n)$	$O(\log n)$ / $O(\log n)$
Heap	$O(1)$ min	$O(n)$ / $O(\log n)$
Stack / Queue	$O(1)$ end	$O(n)$ / $O(1)$

* Amortised: most appends are $O(1)$, an occasional one resizes and costs $O(n)$.