

algo	complexity
LIS	<p>-Size: (binary search)- $O(n \log(n))$.</p> <p>-dynamic programming: $O(n * (n + \log(n))) =$ $= O(n^2 + n \log(n)) \Rightarrow O(n^2)$.</p> <p>-By LCS: $O(n^2) + O(n \log(n))$.</p>
Strategy game	$O((n^2)/2) \Rightarrow O(n^2)$.
MinMax(couple)	$O(n)$.
LCS	$O(m * n) + O(m + n) = O(m * n)$.
Compiler	$O(n) + O(n \log(n)) = O(n \log(n))$.
Power	<p><i>Recursive + Iterative:</i> $O(n)$</p> <p>Improve algorithms: $O(\log(n))$.</p>
Fibonacci	<p>-recursive: $O(2^n)$</p> <p>- iterative: $O(n)$</p> <p>-Improve: $O(\log(n))$.</p>
Max in array	$-O(n)$

2 max in array	$-O(1.5n) \rightarrow O(n)$
Airplane	<i>-dynamic programming: $O(m*n)$</i>
Pizza	$O(1)$
Egg dropping	1 ball – $O(n)$ dividing to equal parts: $O(2 \sqrt{n})$ dividing to different parts: $O(\sqrt{2*n})$
Array matrix	$O(n^2)$
Median	$O(1)$
Donuts problem	$O(1)$