Algorithm:	Complexity:
LIS	***-Size***: (binary search): $O(n * \log n)$ ***-Dynamic programming***: O(n*(n + log(n))) = = O(n^2 + n*log(n)) => $O(n^2)$ ***-By LCS***: $O(n^2) + O(n * \log n)$
Strategy game	O((n^2)/2) => $O(n^2)$
MinMax (couple)	$O(n)$
LCS	O(m*n) + O(m+n) = $O(m * n)$
Compiler	O(n) + O(nlog(n)) = $O(n * \log n)$
Power	*Recursive + Iterative:* $O(n)$ Improve algorithms: $O(\log n)$
Fibonacci	Recursive: $O(2^n)$ Iterative: $O(n)$ Improve: $O(\log n)$
Max in array	$O(n)$

2 max in array	$O(\frac{3}{2}n) => O(n)$
Airplane	*-Dynamic programming*: $O(m * n)$
Pizza	$O(1)$
Glass balls	1 ball: $O(n)$ Dividing equal parts: $O(2 * \sqrt{n})$ Dividing unequal parts: $O(\sqrt{2} * \sqrt{n})$ K balls: $O(K * n^2)$
Array matrix	$O(n^2)$
Median	$O(1)$
Donuts	$O(1)$