



Big O + Time Complexity + Space Complexity

- Time Complexity {
- binary search: check middle element, compare if too big/too small, eliminate 1/2 list. - put in specific order 1st.
 - Dictionary Search: - use keyword to find item. 1 step to find the data.

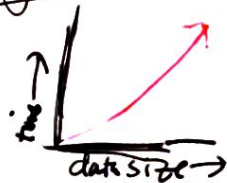
$O(n)$ size of input

best $O(1)$ constant: no Δ with amount of data 

very good $O(\log n)$ logarithmic: complexity ↑ by 1 if double data set. (binary search)
→ usually involve algorithm which executes 1/2 the input size of iteration of a loop.

$O(n)$ linear: If double data size, doubles the complexity. (proportional)


$O(n \log n)$ log linear: input size is multiplied by but 2 log of input size



$O(n^2)$ Quadratic

$O(n^3)$ - n cubed (cubic)

$O(2^n)$ exponential

$O(n!)$ Factorial

complexity is proportional to size of input squared. (often involves 2 nested loops)

complexity doubles each data increase of 1. → recursion (opposite of logarithmic)