

Big O

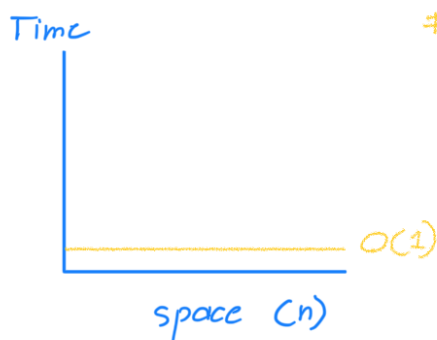
A way to measure how well
an algo performs as the input
grows.

Number of inputs.
↓
 $O(N)$

So as the inputs grow two factors affect the
order of complexity

- Time May take longer to Run
- Space May require more memory to Run

Constant Time Complexity



array

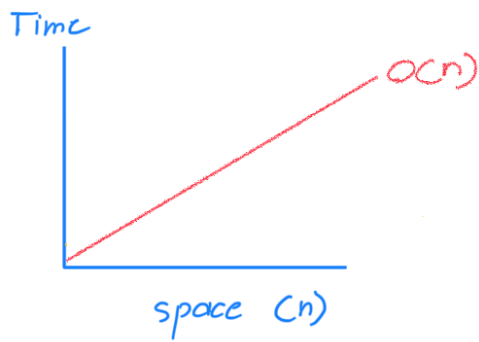
- Pop from End
- push to End
- Lookup

Hashmap

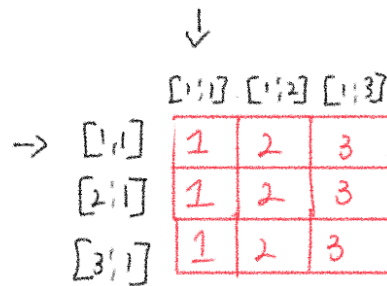
- Insert
- Lookup
- Remove

Linear Growth Complexity → AS Input size grows Time will
grow proportionally.

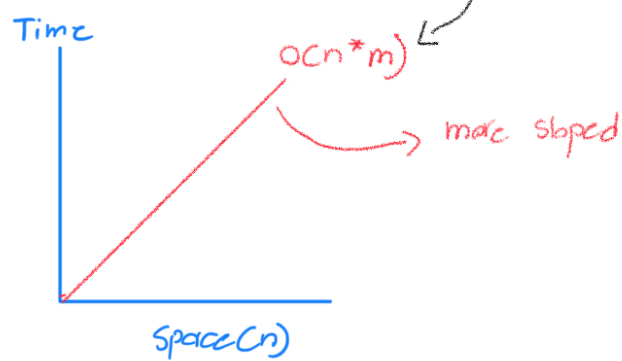
$\underbrace{+1}$ $\underbrace{+1}$ $\underbrace{+1}$



Looping Over elements

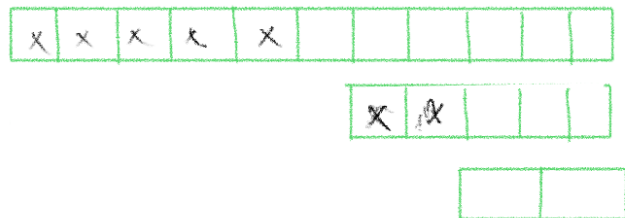
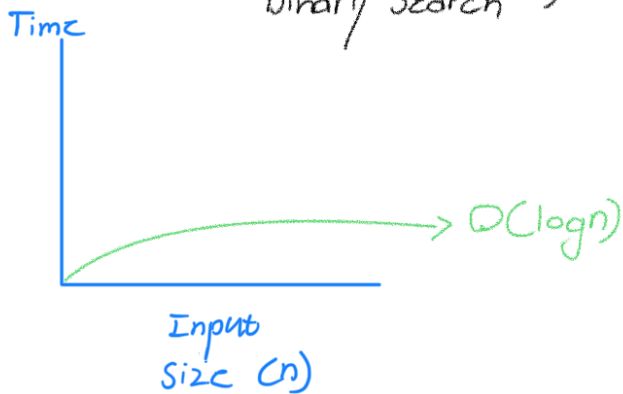


m = Rows
n = Columns



Log N Complexity

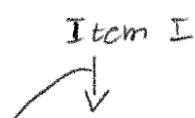
Binary Search →



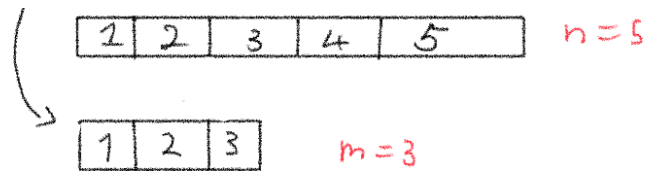
Practice

Nested loops

Inner Loop = 3
Outer Loop = 5



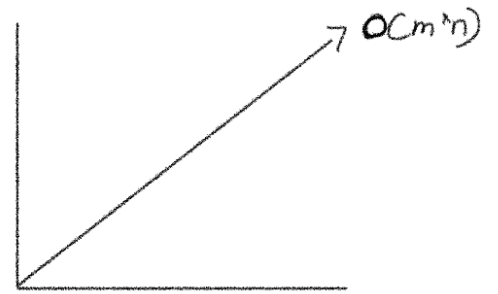
Iterates
over the
nested
array/



$$O(n \times m)$$

$$O(5 \times 3)$$

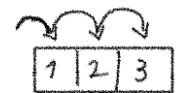
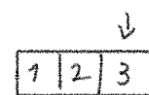
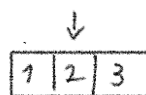
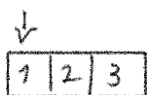
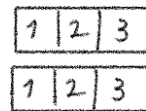
$$O(15)$$



Practice

Nested
Loops

Inner Loop
Outer Loop



Every time you iterate over one element you

$$O(n \times n)$$

$$O(n^2)$$

