# Algorithm Design & Analysis

Case Studies:
- searching
- sorting

Analysis of
- Time
- Space

$\Big\}$ complexity

Revisit Time complexity analysis

1 - Identify critical ops.
2 - Counted # of crit ops.
3 - expressed # of crit. ops. as a function of the size of the input $f(n)$
WC BC AC
4 - $f(n) = \Theta(g(n))$ "Big O notation"

Sol. 1: $f_1(n) = 10{,}000{,}000\, n \log n + n = \Theta(n \log n)$

$f_2(n) = n^2 + 10{,}000{,}000\, n = \Theta(n^2)$

$n \quad n\log n \quad n! \quad n^2 \quad 1 \quad \log n \quad 2^n \quad n^3 \quad n^K \ (K>3) \quad a^n \ (a>2) \quad n^n$

$(\log n)^2 \quad \sqrt{n} = n^{1/2}$

Smallest

$1$
$\log n$
$(\log n)^2$
$\sqrt{n}$
$n$
$n \log n$
$n^2$
$n^3$
$n^K, K>3$
$2^n$
$a^n$
$n!$
$n^n$

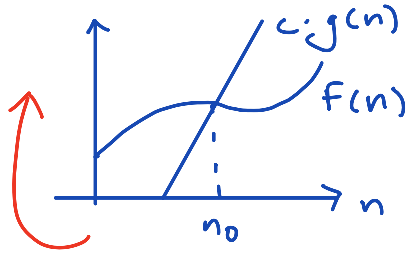
largest

Def: $\Theta$

Asymptotic upper bound

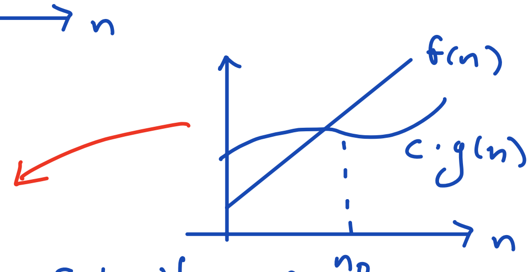
$f(n) = \Theta(g(n))$ if. $\exists c, n_0$ s.t. $\forall n > n_0$

$f(n) \leq c \cdot g(n)$




$\Omega$

Asymptotic lower bound




$f(n) = \Omega(g(n))$ if $\exists c, n_0$ s.t. $\forall n > n_0$

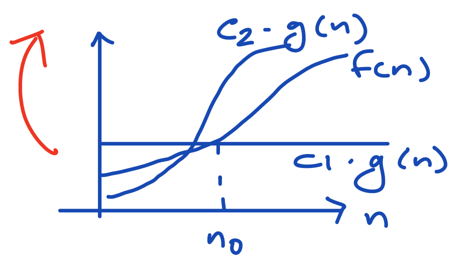
$f(n) \geq c \cdot g(n)$

---

$\Theta$ tight bound

$f(n) = \Theta(g(n))$ if $\exists c_1, c_2, n_0$ s.t. $\forall n > n_0$

$c_1 \cdot g(n) < f(n) < c_2 \cdot g(n)$



# Searching

I. unordered collection of unique items

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 9 & -2 & 4 & -7 & 3 & 5 \end{array} \longleftarrow \text{coll.}$$

$\boxed{\text{Key}} \to$ succ, pos

$\searrow$ unsucc

ex: Key = 4 (succ, 2)
Key = 7 (unsucc)

Methodic Search → Sequential Search [starting @ pos 0]

For every curr item starting at 0

if (curr == Key)
stop → (succ, pos)

else advance

Stop (unsucc)

$$\frac{1+2+3+\dots n}{n} = \frac{\sum_{i=1}^{n} i}{n} = \frac{\frac{\cancel{n}(n+1)}{2}}{\cancel{n}}$$

| | BC | WC | AC |
|---|---|---|---|
| | 0 1 2 | $n-1$ | |
| Succ | 1 2 3 | $n$ | $\frac{n+1}{2}$ |
| unsucc | | $n$ | |

II. Ordered Collection of unique items of size n

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| -7 | -2 | 3 | 4 | 5 | 9 |

for every curr item starting at 0
  if (curr == key)
    stop (succ, pos)

  elseif (curr > key)
    stop (unsucc, pos)

  else
    advance

stop (unsucc, n)

| | Bc 0 | 1 | 2 | . . . . . | Wc n-1 | Ac |
|---|---|---|---|---|---|---|
| succ | 1 | 3 | 5 | 7 | 2n-1 | n |
| unsucc | 2 | 4 | 6 | | 2n | X |

$$\frac{1+3+5+\ldots 2n-1}{n} = \frac{\text{Sum}}{n} = \sum_{i=1}^{2n} i - 2\sum_{i=1}^{n} i = \ldots$$

$$= \frac{n^2}{n} = n$$

Compute Sum

I. add missing terms and then sub

$$\boxed{\begin{aligned} S = 1 + 2 + 3 + 4 + 5 + 6 + 7 + \ldots + 2n-1 + 2n \\ -2 \quad -4 \quad -6 \quad\quad\quad -2n \end{aligned}}$$

$$-2(1+2+3 \ldots + n)$$

$$= \frac{\cancel{2}n(2n+1)}{\cancel{2}} - \cancel{2}\,\frac{n(n+1)}{\cancel{2}} = 2n^2 + \cancel{n} - n^2 - \cancel{n}$$

$$= n^2$$

II. $\sum_{i=1}^{n}(2i-1) = 2\sum_{i=1}^{n} - \sum_{i=1}^{n} 1 = 2\,\frac{n(n+1)}{2} - n$

$= n^2 + n - n = n^2$

III. $\underbrace{1 + \underbrace{3 + 5 + \ldots + 2n}_{} - 3 + 2n - n}_{}$

$\cancel{2}n \times \frac{n}{\cancel{2}} = n^2$

mid index ↓

low, high

[-9 -7 -2] 1 [4 5 9 11]    [0, size-1]

↓ reduce

0 1 2 3 4 5 6 7    [low, high]

$mid = \frac{low + high}{2}$

```
if (Key == mid)
    stop (succ, mid index)

else if (Key < mid)
    high = mid index - 1

else
    low = mid index + 1
```