## Running time analysis (asymptotic analysis)

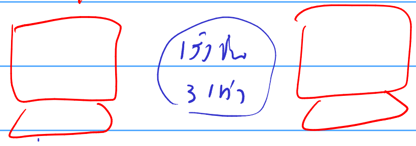
ทราบเวลาการทำงาน

- CPU, mem
- ภาษาโปรแกรม
- input
- OS ....

เวลาการทำงาน เพิ่มได้ เป็น constant เท่า?

Assumptions

"worst case input"




| | $M_1$ | $M_2$ |
|---|---|---|
| n=10 | 1,000 | 3,000 |
| n=100 | 100,000 | ~300,000 |

ต้องการ n ใหญ่

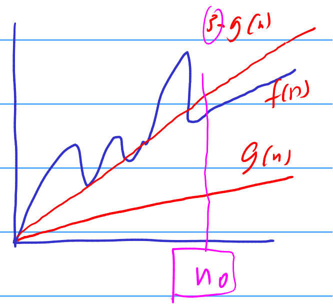
$f(n) = O(g(n))$

### Big-O notation

Definition: จะกล่าวว่า function $f(n)$ เป็น $O(g(n))$ เมื่อ มีค่าคงที่ $C$ และ $n_0$ ที่

$$f(n) \le C \cdot g(n)$$

เมื่อ $n \ge n_0$




$[f(n) = O(g(n))]$

ex $f(n) = 20n^2 + 100n + 500\sqrt{n} = O(n^2)$

พิสูจน์ $20n^2 + 100n + 500\sqrt{n} \le 20n^2 + 100n^2 + 500n^2$

$= 620n^2$ เมื่อ $n \ge 1$

$\omega$

ดังนั้น มี $C = 620$, $n_0 = 1$ ที่ $f(n) \le C \cdot n^2$ $\Rightarrow f(n) = O(n^2)$

▷ พรรคพวก: $O, \Omega, o, \omega, \Theta, \theta$

$o(1)$

▸ จะกล่าวว่า $f(n) = \Omega(g(n))$, ถ้ามี $c, n_0$ ที่

$$f(n) \ge c \cdot g(n) \quad \text{เมื่อ } n \ge n_0$$

$o(n)$

$O(n)$

▸ ถ้า $f(n) = O(g(n))$ และ $f(n)$ เป็น $\Omega(g(n))$, จะเรียกว่า $f(n) = \Theta(g(n))$

## Functions: linear-time

ex. $O(n)$, graph ที่มี $n$ โหนด $m$ edges $O(m+n)$

- merge(A, B)   A & B เป็น list ที่เรียงลำดับแล้ว, output list ที่ merge

  $O(n_1 + n_2)$ เมื่อ $n_1 = |A|$, $n_2 = |B|$

### ▷ Insertion sort

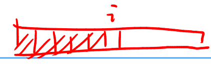
"Loop Invariant"

```
insertionsort(A):
  n ← A.length
  for i ← 1, 2, ... to n-1:
    // insert A[i] to A[0,1,2,...,i-1]
    j ← i-1
    while j ≥ 0
      if A[j] ≤ A[j+1]:
        break
      else swap(A[j], A[j+1])
      j ← j-1
    i ← i+1
```

$1+2+\cdots+n = O(n^2)$

$P_1$: $\boxed{A[0,1,2,\ldots,i-1] \text{ เรียงลำดับแล้ว}}$

$P_1$ ก่อน



ไม่เกิน $O(i)$

$O(1)$

$P_1$ หลัง



Loop invariant P: $\boxed{mx = \max(A[0], A[1], \ldots, A[i-1])}$

```
mx ← A[0], i ← 1
While i < n do
  if A[i] > mx
    mx ← A[i]
  i ← i+1
```

→ จบ loop: $\neg C \wedge P$

**Initialization**: P จริงก่อนเริ่ม

**Preservation / Maintenance**: ถ้า P จริงก่อนทำ body, P จริง หลัง body

**Termination**: เงื่อนไขจบ loop + P

⇒ ผลลัพธ์ที่ต้องการ

---

```
binary search(A, n, target)
  left ← 0, right ← n-1
  while left ≤ right do
    mid ← (left + right) / 2
    if target == A[mid]:
      return mid
    else if target > A[mid]:
      left ← mid+1
    else
      right ← mid-1.
```

ให้ $i^*$ เป็นค่าที่ $A[i^*]$ = target

P: " left $\le i^* \le$ right "

$O(\log n)$

$\log_{a} n = O(\log n)$ เมื่อ a เป็นค่าคงที่

$a > 1,\ b > 1$

$\log_a n = \Theta(\log_b n)$

$$\log_a n = \frac{\log_b n}{\log_b a} = \left(\frac{1}{\log_b a}\right) \log_b n = O(\log_b n)$$

$$n \xrightarrow{1} n - \frac{n}{10} = \frac{9}{10} n \longrightarrow \frac{9^2}{10^2} n$$

ทำไป i รอบ เหลือ $\le \left(\frac{9}{10}\right)^i n$ เมื่อใด $\left(\frac{9}{10}\right)^i n < 1$

$$n < \left(\frac{10}{9}\right)^i$$

$$i > \log_{\frac{10}{9}} n = O(\log n)$$

---

หา primes $1, \dots, N$. — Sieve

```
p[2] ← true, p[0..] ← .
for i ← 2, ..., N
    if p[i]:
        j ← i × 2
        while j ≤ N
            p[j] ← false , j ← j + i .
```

เวลาการทำงาน

$$\le \frac{N}{2} + \frac{N}{3} + \frac{N}{4} + \frac{N}{5} + \frac{N}{6} + \dots + \frac{N}{N}$$

$$\le N \underbrace{\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}\right)}$$

upperbound

$$1 + \overbrace{\left(\frac{1}{2} + \frac{1}{3}\right)}^{2} + \overbrace{\left(\frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}\right)}^{4} + \overbrace{\left(\frac{1}{8} + \dots\right)}^{8} \quad \overbrace{(\quad)}^{16}$$

$$\le 1 + 2 \cdot \frac{1}{2} + 4 \cdot \frac{1}{4} + 8 \cdot \frac{1}{8} + 16 \cdot \frac{1}{16} \qquad \le 1 + \log_2 n$$

lower bound

$$1 + \boxed{\frac{1}{2}} + \boxed{\frac{1}{3} + \frac{1}{4}} + \boxed{\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8}} + \dots$$

$$\ge 1 + 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 4 \cdot \frac{1}{8} + \dots \qquad \ge \frac{1}{2} + \frac{1}{2} + \dots \ge \frac{\log_2 n}{2}$$

## Geometric series

$$n+\frac{n}{2}+\frac{n}{4}+\frac{n}{8}+\frac{n}{16}+\frac{n}{32}+\cdots = \frac{n}{2^0}+\frac{n}{2^1}+\frac{n}{2^2}+\cdots$$

$$= n\Big(\underbrace{1+\tfrac{1}{2}+\tfrac{1}{4}+\tfrac{1}{8}+\cdots}_{\le 2}\Big)$$

$$\le O(n)$$

$$a^0+a^1+a^2+a^3+\cdots = \frac{1}{1-a} \quad \text{ถ้า } a<1$$

---

$n = 2^{e}$

```
while n > 1
    n ← √n
```

```
k = 2
while k < n
    k ← k²
```

$\boxed{\log\log n}$

---

<u>exponential-time</u> $2^n$, $c^n$, $1.75^n$ $2^{n/5}$

รับ n , n เป็น จน.เฉพาะ หรือไม่ ?

$O(\sqrt{n})$

```
for i ← 2, 3, ..., √n
    if n mod i == 0
        return false
return true.
```

→ gcd (a, b): assume $a \ge b$

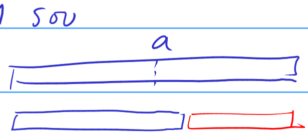
```
    if a mod b == 0 , return b
    else return gcd(b, a mod b)
```

gcd(100, 51) ↓ gcd(51, 49) ↓ gcd(49, _)

gcd(100, 97) ↓ gcd(97, 3) ↓ gcd(3, _)

Case 1: ถ้า $b \le a/2$ → จบ 1 รอบ

Case 2: ถ้า $b > a/2$



$a \bmod b = a-b \le a/2$

gcd($\boxed{a}$, b)

gcd (b, a mod b)

gcd ($\boxed{a \bmod b}$, __ )