Lecture 20, 23 October 2025

Madhavan Mukund

`https://www.cmi.ac.in/~madhavan`

Programming and Data Structures with Python

Lecture 20, 23 Oct 2025

Algorithms + Data Structures = Programs

Niklaus Wirth

Pascal

Efficiency

Resources

Time

Space

"Classical" memory based computation

"Ignore" space

Example

Checking for common elements between 2 lists

```
common = []
for x in l1:        n      n
    for y in l2:    m      n
        if x == y:  ← n×m times    n^2
            common.append(x)
```

Application

Check if every SIM card registration has a valid Aadhaar number

n SIMlist

m Alist

Check common elements

Earlier nested loop $= |\text{SIMlist}| \times |\text{Alist}|$

$n \cdot m$

What are n & m?

m? $\approx$ pop of India $\simeq$ 140 crore $\approx 10^9$

n? Also 10^9

Nested loop takes $10^9 \times 10^9 = 10^{18}$ iterations

At least 10^{18} "basic operations"

Processing capacity

Python = approx 10^7 ops/sec

C/C++ = upto 10^8 or 10^9 ops/sec

10^{18} ops $\xrightarrow{\text{Python}}$ 10^{11} seconds $10^{18}/10^{7}$

$10^{11}/60 \approx 1.6 \times 10^{9}$ minutes

$1.6 \times 10^{9}/60 \simeq 1.6 \times 10^{7}$ hours

$1.6 \times 10^{7}/24 = 7 \times 10^{5}$ days

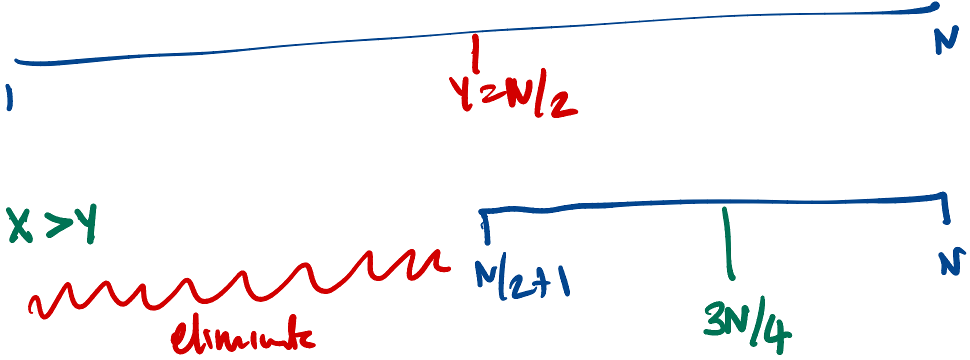
$7 \times 10^{5}/365 = 1.x \times 10^{3}$ years

Solution

Think of X in $[1,N]$. Guess it?

Make a guess Y,
- Yes $X = Y$
- No $X < Y$
- No $X > Y$

Strategy – start with $Y = N/2$ – Guaranteed to halve the search space



$$N \to N/2 \to N/4 \;\ldots\; \to 1 \qquad \log_2 N$$

4 2

?

Sort AList

for x in SIMList $\leftarrow n$

Search for x in sorted Alist $\leftarrow \log n$ — "Binary search"

$n^2 \longrightarrow n \log n$

$10+10+10$

$10^3 \times 10^3 \times 10^3$

$2^{10} = 1024 \simeq 10^3$

$\log_2 10^3 = 10$

$10^9 \times 10^9$

$10^9 \times \log_2 10^9 \simeq 30$

10^{18} ops $\longrightarrow$ 30×10^{9} ops

$\downarrow$

30×10^{2} secs $=$ 3000 secs

$=$ 50 minutes

$\approx$ 1000s years $\longrightarrow$ 50 minutes

Requires efficient sorting of Alist

Analysis of algorithms

What are we measuring (calculating) Time

Depends on input size

$T(n)$ - time on input of size n

Which input of size n?

- Ideally "average" case
- Settle for "worst" case

Input size?

Structure — list : no. of elements

Number theoretic algorithm

Addition of multidigit numbers

$$
\begin{array}{r}
{}^{1}174 \\
+\,892 \\
\hline
1\,0_3\,6_2\,6_1
\end{array}
$$

$\xrightarrow{10\times\text{ value}}$ +1 steps

1745
8928

# of digits of n in base 10 = ? $\simeq \log_{10} n$

Primality

isprime(n)?

Check divisibility by 2 .. n−1 $\simeq$ n steps

$\downarrow$

$10^{\log_{10} n}$

Asymptotic Worst Case Complexity

Orders of magnitude

$\frac{n(n+1)}{2}$ is essentially n^2

Why is $\frac{n^2}{2} + \frac{n}{2}$ "same" as n^2?

$\exists c$ s.t. $\frac{n^2}{2} + \frac{n}{2} < c \cdot n^2$ for every n

n^2+n is "same as" n^2

$$n^2+\underline{n} \leq n^2+\underline{n^2}$$

$$\leq 2n^2 \text{ for all } n$$

"Focus on largest term"

$\underline{\exists c}$ $f(n) \leq c\,g(n)$ for all n

n^2+n $\quad$ n^2

$f(n)$ is $O(g(n))$

$n^k = O(n^\ell)$ if $k \leq \ell$

$\log n \quad \sqrt{n} \quad n \;\|\; n^2 \quad \ldots \quad 2^n$

Orders of magnitude

Input size	Values of $t(n)$						
	$\log n$	n	$n \log n$	n^2	n^3	2^n	$n!$
10	3.3	10	33	100	1000	1000	10^6
100	6.6	100	66	10^4	10^6	10^{30}	10^{157}
1000	10	1000	10^4	10^6	10^9		
10^4	13	10^4	10^5	10^8	10^{12}		
10^5	17	10^5	10^6	10^{10}			
10^6	20	10^6	10^7	10^{12}			
10^7	23	10^7	10^8				
10^8	27	10^8	10^9				
10^9	30	10^9	10^{10}				
10^{10}	33	10^{10}	10^{11}				