R Reading P20 Pseudocode
2-2 Analyzing Alg
Lectuer 1

Sep 30, 2018 Pd 1

Part I

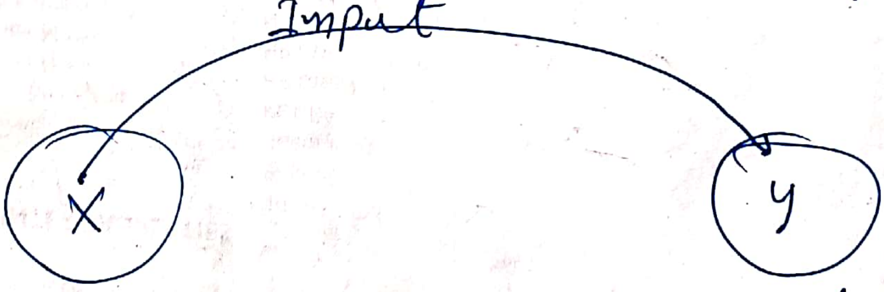
Key notion ;

Algorithm = well - specified procedure for solving a computional problem

( Mathematical abstraction of Computer Progr

→ May be specified in English (preferred) or pesudo-code, as long as it is precise (Avoid using real code)

problem specifies desired output for each Input




$y = f(x)$

input ("Instances")

outputs

Sep 30, 2018

e.g

X = an integer
y = smallest prime $\geq x$

→ Difference between cs and math we care how f is computed

Want algorithms that are:

→ Correct (obviously!)
→ fast ≃ scalable
→ Simple

focus here

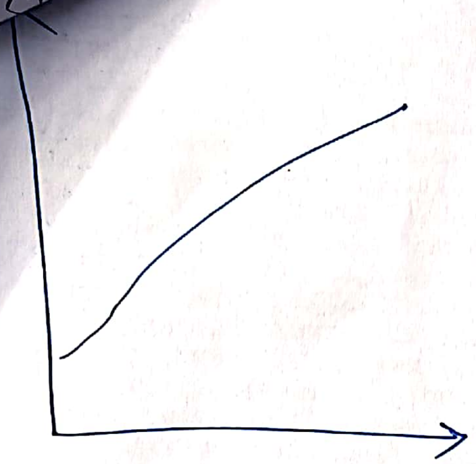
Scalability :

→ Measure running time / space / etc as input size grows
→ our focus : $T(n)$ = run time as function of input size n

need to be defined for each problem
e.g. n for n×n matrix input

Scanned with CamScanner

Sep 30, 2018 P.3




→ we care only about "big picture" here

→ Ignore minor details (machine instructions set, Compiler optimization, ...)

very important idea: Ignore costant factors and lower orders terms

→ Key tools: Asymptotic analysis ($\Theta$, $O$, $\Omega$, $o$, $\omega$)

E.g $5n^2 - 7n + 4 = \Theta(n^2)$

Scanned with CamScanner

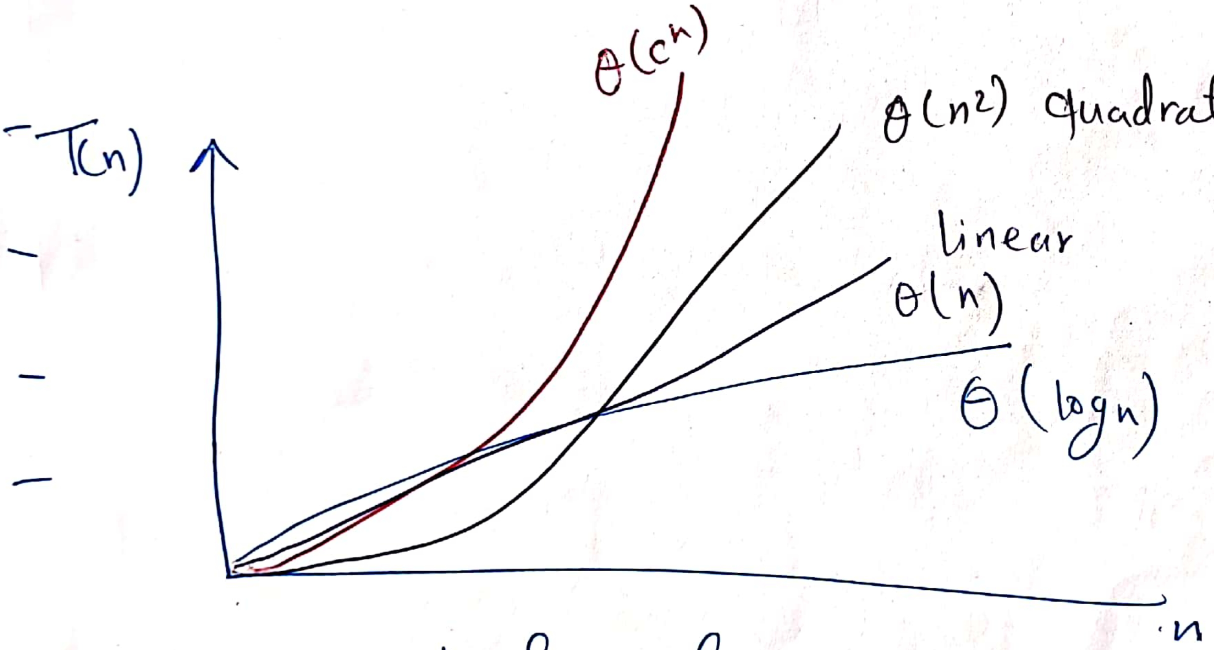
Sep 30, 2018

$T(n) = \Theta(\log n)$ logorithmic } GREAT

$T(n) = \Theta(n)$ linear }

$T(n) = \Theta(n^2)$ quadratic Ok

$T(n) = \Theta(c^n)$ exponential BAD




→ our chief goal here =

Learn how to reason about correctness and efficiency of algorithms in precise and principle manner

Scanned with CamScanner

(P 5)

Sep 30, 2018

Material overview:

→ Sorting (one of the most basic problem)

→ Data structures (organizing data to make it easy to access)

→ Heaps

→ Binary search trees

→ Hashing

→ graph search (how to explore graph)

→ shortest path (how Google Maps works)

→ iterative algorithms (optimization via repeated

→ Dynamic programming

Scanned with CamScanner

Basic idea :



Mathematical model of Computer. Mentally execute algorithm on model and evaluate time

- What is the time required on the model for every operations that an algorithm might Perform
- what input data

$Q_1$ : How does our model relate to real Computer?

I will discuss some definition

Problem we give defintion for it above let give ex

- Computing GCD greatest Common divisor
- Finding shortest path on map
- Finding meaning of a word in dictionary
- Given x ray determin if there is disease

GCD of two number our Input Consist of number 36, 48 the GCD be number 12

Scanned with CamScanner

Sep 30, 2018

Problem: Specification of what are valid input and what are acceptable output for each valid Input

Input Instance: A value x is an input instance for problem P, if x is a valid input as per the Specification

So 36, 48 Constitute an instance for GCD

Another important term that we need is size of instance

Algorithm: An abstract Computational procedure which takes som value or values as input and produces a value or values as output

program = Expression of Algorithm

Mathematical model that we are going to use in this Course is called RAM: stands for random access machine. This is a very simplified computer mode Consist of two parts

1. processors
2. Memory

Scanned with CamScanner

- How to write Algorithem Pesudocode See Convention
- Frequency Count Method

Algorithm Sum (A, n)    // A is array
// n is size of element

```
{
    S=0  ——————————  1
    for (i=0; i<n; i++) — n+1
         1    n+1  n
    {
        S = S + A[i];  ——  n
    }
    return S  ——————  1
}
```

A

| 8 | 3 | 9 | 7 | 2 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |

n = 5
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5 X stop

$f(n) = 2n + 3$ this time funct

Assume assign each statement with unit of time

the degree is n so order of (n)

Space analysis

$A \longrightarrow n$ words
$n \longrightarrow 1$
$S \longrightarrow 1$
$i \longrightarrow 1$

Space Complexity $S(n) = n + 3$ is $O(n)$

Scanned with CamScanner

ex2

Algorithm add (A, B, n) // A, B are squar matrix
{
for(i=0; i<n, i++) — n+1
{
for(j=0; j<n; j++) — n x (n+1)
{
c[i,j] = A[i,j] + B[i,j]; — nxn
}
}
}

$f(n) = 2n^n + 2n + 1$

nxn matrix 3x3

degree of $n^2$ so Order of $n^2$ $O(n^2)$

Space complexity

A $n^2$ nxn
B $n^2$
C — $n^2$
n — 1
i — 1
j — 1

$S(n) = 3n^2 + 3$ $O(n^2)$

Scanned with CamScanner

Algorithm multiply ( A, B, n)

```
{
    for ( i=0; i<n ; i++) ———— n+1
    {
        for (j = 0; j<n; j++) ——— n*(n+1)
        {
            C[i, j] = 0; ———— n x n
            for( k=0; k<n; k++) —— n xn x(n+1)
            {
                c [i, j] += A[i,k] * B[k, j]; —— n x n x n
            }
        }
    }
}
```

$f(n) = 2n^3 + 3n^2 + 2n + 1$

$O(n^3)$ Time

$O(n^3)$ order of Cub

Space

| | | |
|---|---|---|
| A | $n^2$ | |
| B | $n^2$ | $S(n) = 3n^2 + 4$ |
| C | $n^2$ | order of $n^2$ |
| n | 1 | |
| i | 1 | |
| j | 1 | |
| K | 1 | |

```
for (i=0; i<n, i++)      n+1
{
    stat  ———  n
}
```

$$O(n)$$

if suppos

رغم انها تنقص

```
for (i=n; i>0, i--)
{
    stat  ————  n
}
```

$$O(n)$$

if we chang the loop

How to analyze one

```
for (i=1; i<n, i=i+2)
{
    stat  ————  n/2
}
```

$$O(n)$$

if chang

```
for (i=1; i<n; i=i+20
{   stat  ————  n/20
}
```

$$O(n)$$

رغم order of n

Scanned with CamScanner

```
for ( i=0 ; i<n , i++)        n+1
{
    for (j=0; j<n ,j++)       n x (n+1)
    {
        stat ________________ n x n
    }
}
```

$O(n^2)$

Another Loop

```
for (i=0; i<n; i++)
{
    for (j=0 ; j<i , j++)
    {
        stat
    }
}
```

0 < 0 no

عدد المرات التنفيذ

Trace to analyze

| i | j | # of times |
|---|---|---|
| 0 | 0x | 0 |
| 1 | 0 exc<br>1x | 1 |
| 2 | 0<br>1<br>2x | 2 |
| 3 | 0<br>1<br>2<br>3x | 3 |
| ⋮ | | ⋮ |
| up to n | | n |

$1+2+3+\dots+n$

$= \frac{n(n+1)}{2}$

$f(n) = \frac{n^2+1}{2}$

$f(n) = O(n^2$

P. 13

```
P = 0
for( i=1 ; p <= n, i++)
{
      P = P + i
}
```

Trace

| i | p |
|---|---|
| 1 | 0+1=1 |
| 2 | 1+2=3 |
| 3 | 1+2+3 |
| 4 | |
| ⋮ | |
| k times | |

$1+2+3+4+\dots+k$

Assume becam $P > n$

since $P = \frac{k(k+1)}{2}$

$\frac{k(k+1)}{2} > n$

$k^2 > n$

$k = \sqrt{n}$

this algorithm order of $O(\sqrt{n})$

Scanned with CamScanner

```
for (i=1; i<n; i=i*2)
{
    stat
}
```

| i |
| --- |
| 1 |
| 1x2 = 2 |
| 2x2 = 4 |
| 4 x 2 = 8 |
| ⋮ |
| $2^k$ |

Assume i becam i > n

$i = 2^k$

$2^k >= n$

$2^k = n \longrightarrow k = \log_2 n$

$O(\log_2 n)$

حلو كتابة لهذا التكرار العدد for

```
for (i=1 ; i<=n , i++)
{    stat
}
```

$i = \underbrace{1+1+1+\dots}_{k} = n$

in for(i=1, i<n, i=i*2

$i = 1 \times 2 \times 2 \times \dots = n$

$2^k = n$

$k = \log_2 n$

Suppos $n = 8$ if take $n = 10$

Trace

| $i$ | |
|---|---|
| 1 | |
| 2 | 3 |
| 4 | |
| 8x | |

| $i$ | |
|---|---|
| 1 < 10 | |
| 2 | |
| 4 | 4 |
| 8 | |
| 16x | |

if find $\log 8 = 3$ $\log_2 2^3 = 3 \cancel{\log_2 2}$ 1

ceil valu $\log_n 10 = 3.2$

$\lceil \log n \rceil$

another Code

```
for (i=n, i>=1; i=i/2)
{
    stat
}
```

Trace $i$

| $i$ | |
|---|---|
| $n$ | start |
| $\frac{n}{2}$ | |
| $\frac{n}{2^2}$ | |
| ⋮ | |
| $\frac{n}{2^k}$ | |

Assume $i < 1$

$\frac{n}{2^k} < 1$

$\frac{n}{2^k} = 1$

$2^k = n$

$k = \log_2 n$

$O(\log_2 n)$

Scanned with CamScanner

```
for(i=0; i*i<n, i++)
{
    stat
}
```

Terminat

$i*i \geq n$

$i^2 = n$

$i = \sqrt{n}$

So order of $O(\sqrt{n})$

---

Ex

```
for (i=0; i<n, i++)
{
    stmt  ____________  n
}
for (j=0; j<n, j++)            Not nested
{
    stmt)  ___________  n
}
```

$\overline{\phantom{xxxxxx}2n}$

$O(n)$

Next one

```
p=0
for( i=1 ; i<n ; i=i*2
{
    p++                    log n
}
for ( j=1 ; j<p ; j=j*2 )
{
    stmt  ____________   log(p)   Log n
}
```

شرح تجار الطلاب

Second loop is log p
Where p = log n
So loglogn

$\longrightarrow \log \log n$

$O(\log \log n)$

what is p
p in loop befor

```
for (i=0; i<n, i++)  ——— n
{
    for (j=1 ; j<n, j=j*2)  ——— n × log n
    {
        stmt
    }                       ——— n × log n
}
```

$2n\log n + n$

$O(n\log n)$

Summary

for (i=0, i<n, i++) — $O(n)$

for (i=0 ; i<n, i=i+2)  $\frac{n}{2}$  $O(n)$

for (i=n; i>1, i--) — $O(n)$

for (i=1 ; i<n, i=i×2) — $O(\log_2 n)$

for (i=1; i<n, i=i×3) — $O(\log_3 n)$

for (i=n, i>1; i=i/2) — $O(\log n)$

# Analysis while & if

```
while (Condition)
{
    stmt;
}
```

ex

```
i = 0                 ———— 1
while (i < n)         ———— n + 1
{
    stmt;             ———— n
    i++;              ———— n
}
```

$f(n) = 3n + 2 \Rightarrow O(n)$

ex

```
a = 1
while (a < b)
{
    stmt
    a = a*2
}
```

Trace

| a | |
|---|---|
| 1 | |
| $1 \times 2 = 2$ | $2$ |
| $2 \times 2 = 4$ | $2^2$ |
| $4 \times 2 = 8$ | $2^3$ |
| ⋮ | |
| $2^k$ | |

stopping when $a \geq b$

~~$2^k$~~ $\quad a = 2^k$

$2^k \geq b$

$k = \log_2 b$ replace b with n

$O(\log_2 n)$

ex

```
i = n
while (i > 1)
{
    stmt;
    i = i/2;
}
```

ex

```
r = 1
k = 1
while (k < n)
{
    stmt
    k = k + i
    i++;
}
```

| i | k |
|---|---|
| 1 | 1 |
| 2 | 1+1 = 2 |
| 3 | 2 + 2 = |
| 4 | 2+2+3 |
| 5 | 2 + 2+3+4 |
| m | |

Let m times

$\cancel{2}^{1} + 2 + 3 + 4 + \dots + m = \frac{m(m+1)}{2}$

$k > n$ will stop

$\frac{m(m+1)}{2} \geq n$

$m^2 \geq n \Rightarrow m = \sqrt{n}$ $\quad O(\sqrt{n})$

```
while (m != n)
{
    if (m > n)
        m = m - n
    else
        n = n - m
}
```

P. 19

| m | n | |
|---|---|---|
| 6 | 3 | |
| 3 | 3 | become equal only one time |
| 5 | 5 | not executed 0 times |

| m | n |
|---|---|
| 16 | 2 |
| 14 | 2 |
| 12 | 2 |
| 10 | 2 |
| 8 | 2 |
| 6 | 2 |
| 4 | 2 |
| 2 | 2 |

7 times, almost $\frac{16}{2}$

$\frac{n}{2} \rightarrow O(n)$

min $O(1)$

max $O(m)$

## Types of time function

$O(1)$ —— Constant

$f(n) = 2 \rightarrow O(1)$

$f(n) = 5 \rightarrow O(1)$

$f(n = 500 \rightarrow O(1)$

$O(\log n)$ Logrithemic

$O(n)$ Linear

$f(n) = 2n + 3 \rightarrow O(n)$

$f(n)\ 500n + 700 \rightarrow O(n)$

$O(n2)$ quadratic

$O(n3)$ cubic

$O(2^n)$ Exponential

$O(3^n)$

$O(n^n)$

Scanned with CamScanner

$1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 < \dots < 2^n < 3^n < n^n$

P. 20

To prove

logi = 0

$\log_2$

| $\log n$ | $n$ | $n^2$ | $2^n$ |
|---|---|---|---|
| 0 | | | |
| 1 | 1 | 1 | 2 |
| 2 | 2 | 4 | 4 |
| 3 | 4 | 16 | 16 |
| 3.1 | 8 | 64 | 256 |
| | 9 | 81 | 512 |

Scanned with CamScanner