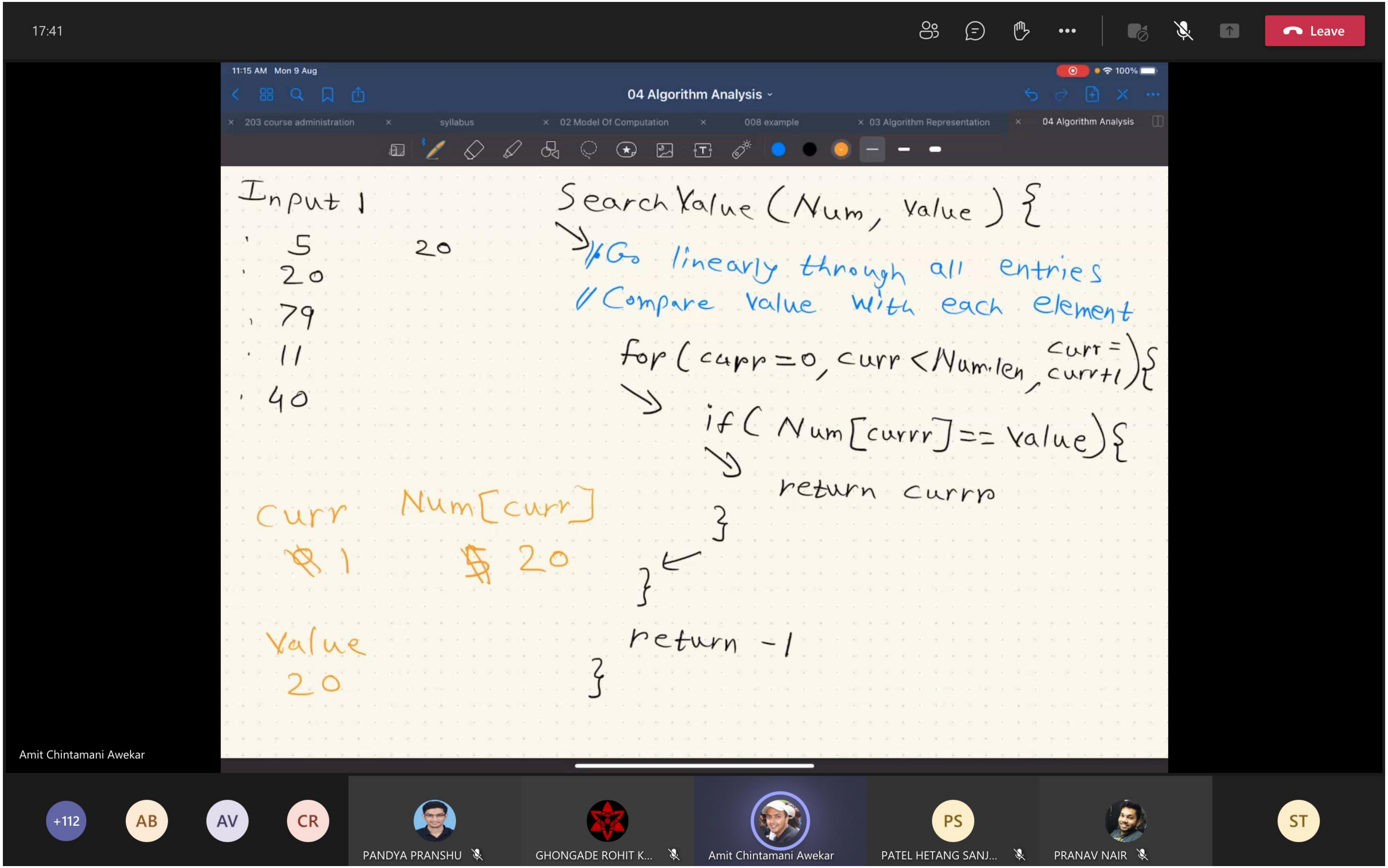

17:41
Leave
11:15 AM Mon 9 Aug
100%
04 Algorithm Analysis
203 course administration
syllabus
02 Model Of Computation
008 example
03 Algorithm Representation
04 Algorithm Analysis
Input
5
20
79
11
40
20
SearchValue (Num, Value) {
// Go linearly through all entries
// Compare value with each element
for (curr = 0, curr < Num.len, curr = curr+1) {
if (Num[curr] == value) {
return curr
}
}
return -1
}
curr
~~0~~ 1
Num[curr]
~~5~~ 20
Value
20
Amit Chintamani Awekar
+112
AB
AV
CR
PANDYA PRANSHU
GHONGADE ROHIT K...
Amit Chintamani Awekar
PS
PATEL HETANG SANJ...
PRANAV NAIR
ST

45:15

11:42 AM Mon 9 Aug

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

How to covert relation to function

- Consider only the worst case
- Average case
- Best case

Value absent

Value present at the end

Flight

Industrial plant

Amit Chintamani Awekar

+112 VG AM Amit Chintamani Awekar PANDYA PRANSHU GHONGADE ROHIT K... AADI AARYA CHANDRA VIRAJ GOYANKA ST

$f(n) \in \Theta(g(n))$

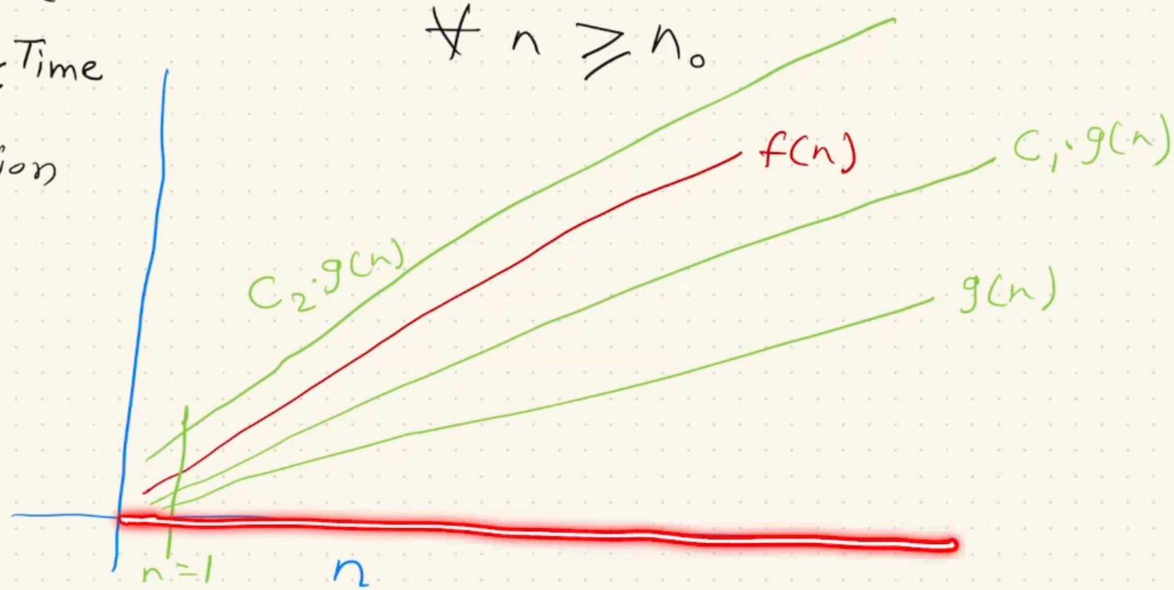
$c_1 \quad c_2 \quad n_0$

Maps problem size to resource consumption

Time

$0 \leqslant c_1 \cdot g(n) \leqslant f(n) \leqslant c_2 \cdot g(n)$

$\forall n \geqslant n_0$



58:20 Leave

11:55 AM Mon 9 Aug 94%

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

$f(n) = n^2 + 2n + 5$ $g(n) = n^2$ $c_1 = 1$ $c_2 = 8$ $n_0 = 1$

n	$f(n)$	$g(n)$	$c_1 \cdot g(n)$	$c_2 \cdot g(n)$
0	5	0	0	0
1	8	1	1	8
2	13	4	4	32
3	20	9	9	72
4	29	16	16	128
5	40	25	25	200

$f(n) \in \Theta(g(n))$ $c_1 = 1$ $c_2 = 8$ $n_0 = 1$

Amit Chintamani Awekar

+112 AC AADI AARYA CHANDRA | Amit Chintamani Awekar | VG VIRAJ GOYANKA | KS KANDULA SRAVYA | AC ABHINABA CHAKRAB... | ST

13:50 Leave

2:12 PM Mon 9 Aug 100%

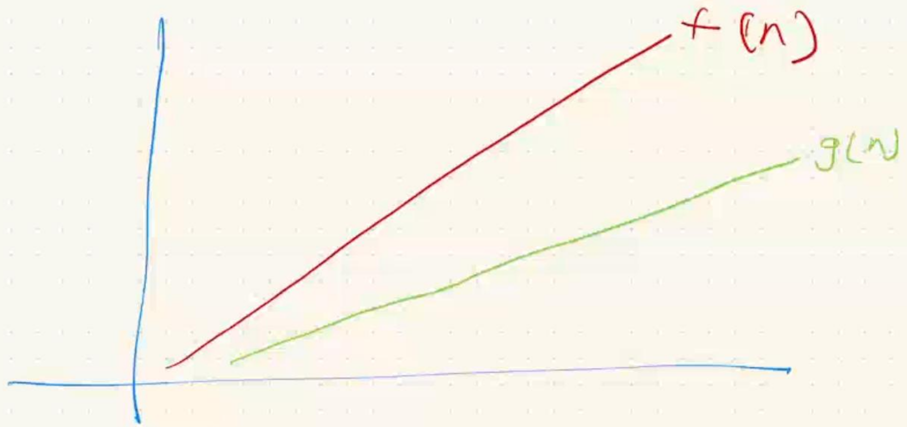
04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

$f(n) \in O(g(n))$

$c \quad n_0$

$$0 \leq f(n) \leq c \cdot g(n) \quad \forall n \geq n_0$$



Amit Chintamani Awekar

+113 GR ABHISHEK KUMAR M MANSI Amit Chintamani Awekar YG YASH GARG PRANAV NAIR ST

21:16

Leave

2:19 PM Mon 9 Aug

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

$f(n) = n^2 + 5n + 2 \qquad g(n) = n^2 \qquad f(n) \in O(g(n))\ ?$

$c \qquad n_0$

$f(n) = 5n + 2 \qquad g(n) = n^2 \qquad c = 2 \qquad n_0 = 3$

n	$5n+2$	n^2	$2 \cdot n^2$	$0.5\,n^2$
0	2	0	0	0
1	7	1	2	0.5
2	12	4	8	2
3	17	9	18	4.5
4	22	16	32	8
5	27	25	50	12.5
6	32	36		18

$c = 0.5 \qquad n_0 =$

Amit Chintamani Awekar

+113 GR M Amit Chintamani Awekar YASH GARG PRANSHU PRANAV NAIR VATSAL GUPTA ST

23:58

2:21 PM Mon 9 Aug

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

Θ	O	Ω
asymptotic tight bound	asymptotic upper bound	asymptotic lower bound
$=$	$\leq$	$\geq$

Amit Chintamani Awekar

+112 GR M Amit Chintamani Awekar YASH GARG PRANSHU PRANAV NAIR VATSAL GUPTA ST

28:20

Leave

2:26 PM Mon 9 Aug

100%

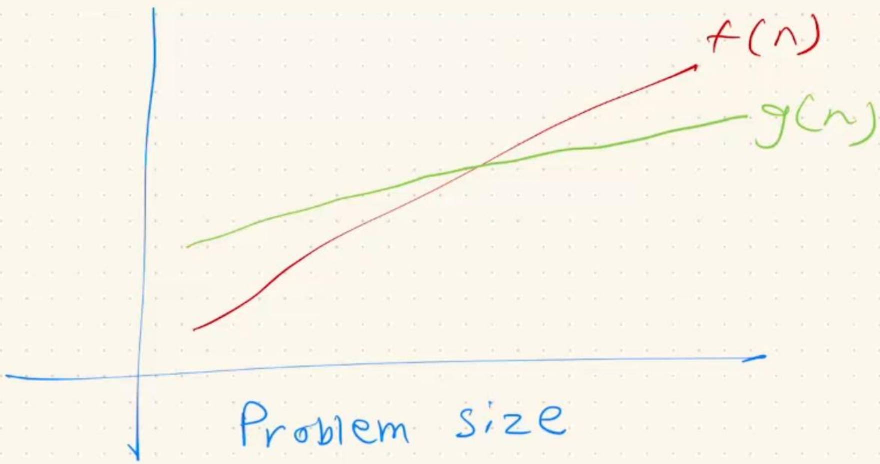
04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

$$f(n) \in \Omega(g(n))$$

$$c \quad n_0$$

$$0 \leq c \cdot g(n) \leq f(n) \quad \forall \; n \geq n_0$$



Amit Chintamani Awekar

+111 M YG VG Amit Chintamani Awekar PRANSHU PRANAV NAIR GUNJAN DHANUKA ANINDYA VIJAYVARG... ST

32:46

Leave

2:30 PM Mon 9 Aug

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

$f(n) = n^2 + 5n + 2 \quad g(n) = n^2 \quad f(n) \in \Omega(g(n))$?

$c = 1 \quad n_0 = 0$

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

$c = 500$

12 of 12

Amit Chintamani Awekar

+112 VG Amit Chintamani Awekar AV ANINDYA VIJAYVARG... PK PRANSHU KANDOI VK VISHAL KUMAR PRANAV NAIR ST

04 Algorithm Analysis

Transitivity:

$A : B, \quad B : C \longrightarrow A : C$

$f(n) \in \Theta(g(n)), \quad g(n) \in \Theta(h(n)) \longrightarrow f(n) \in \Theta(h(n))$

$\quad O \qquad\qquad\qquad O \qquad\qquad\qquad O$

$\quad \Omega \qquad\qquad\qquad \Omega \qquad\qquad\qquad \Omega$

35:31

Leave

2:33 PM Mon 9 Aug

99%

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

Reflexivity:

A : A

$f(n) \in \Theta(f(n))$

$f(n) \in O(f(n))$

$f(n) \in \Omega(f(n))$

Amit Chintamani Awekar

+113 VG Amit Chintamani Awekar ANINDYA VIJAYVARG... PRANSHU KANDOI VISHAL KUMAR PRANAV NAIR ST

38:13 Leave

2:36 PM Mon 9 Aug

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

Symmetry:

$$A ; B \longleftrightarrow B ; A$$

$$f(n) \in \Theta(g(n)) \longleftrightarrow g(n) \in \Theta(f(n))$$

$$\downarrow \qquad \qquad \qquad \qquad \uparrow$$

$$0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \quad \forall n \geq n_0$$

$$g(n) \leq \frac{f(n)}{c_1} \qquad \frac{f(n)}{c_2} \leq g(n) \quad \forall n \geq n_0$$

$$1/c_1 = c_1' \qquad 1/c_2 = c_2'$$

Amit Chintamani Awekar

+112 AV Amit Chintamani Awekar PK PRANSHU KANDOI VK VISHAL KUMAR PRANAV NAIR AG AYAN GAUTAM ST

41:04

Leave

2:38 PM Mon 9 Aug

98%

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

$$f(n) \in O(g(n)) \longleftrightarrow g(n) \in O(f(n))$$

$$\Omega \qquad\qquad\qquad \Omega$$

Counter example

Amit Chintamani Awekar

+111 AV VK Amit Chintamani Awekar PK PRANSHU KANDOI PRANAV NAIR AG AYAN GAUTAM AC AADI AARYA CHANDRA ST

48:08

Leave

2:46 PM Mon 9 Aug

97%

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

Fibonacci Numbers

$fib(1) = 0 \quad fib(2) = 1 \quad fib(n) = fib(n-1) + fib(n-2)$

FibR(n)

```
if(n==1){
    return 0
}
if (n== 2){
    return 1
}
answer=FibR(n-1) + FibR(n-2)
return (answer)
```

$T(n) = T(n-1) + T(n-2)$

$T(n) \simeq (1.6)^n$

n	Fib(n)
1	0
2	1
3	1
4	2
5	3
6	5
7	8
8	13

Amit Chintamani Awekar

+111 AG AC

Amit Chintamani Awekar

YG YASH GARG

NS NANDINI SHARMA

VG VATSAL GUPTA

PK PRANSHU KANDOI

ST

58:51

Leave

2:57 PM Mon 9 Aug

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

```
FibS( n){
    if(n==1){
        return 0
    }
    if (n== 2){
        return 1
    }
    fibn1 = 1      fib(n-1)
    fibn2 = 0      fib(n-2)
    for(curr=3, curr<=n, curr=curr+1){
        answer= fibn1 + fibn2
        fibn2 = fibn1
    }   fibn1 = answer
```

To compute n^{th} number

Start from 3^{rd} number

and then compute 4^{th}

then 5^{th}

⋮

till you reach n^{th}

Amit Chintamani Awekar

+112 YG VG Amit Chintamani Awekar NS NANDINI SHARMA PK PRANSHU KANDOI PRANAV NAIR KH KILAPARTHI HEEMAN... ST

```
FibS( n){
    if(n==1){
        return 0
    }
    if (n== 2){
        return 1
    }
    fibn1 = 1        <- fib(n-1)
    fibn2 = 0        <- fib (n-2)
    for(curr=3, curr ≤ n, curr=curr+1){
        answer = fibn1 + fibn2
        fibn2 = fibn1
        fibn1 = answer
    }
    return answer
```

To compute n^{th} number

Start from 3^{rd} number

and then compute 4^{th}

then 5^{th}

⋮

till you reach n^{th}

01:20:53

Leave

3:18 PM Mon 9 Aug

90%

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

```
FibS ( n) {
    if(n==1){
        return 0
    }
    if (n== 2){
        return 1
    }
    fibn1 = 1
    fibn2 = 0
    for(curr=3, curr≤ n, curr=curr+1){
        answer= fibn1 + fibn2
        fibn2 = fibn1
        fibn1 = answer
    }
    return answer
}
```

c_1

$c_2 \cdot n$

c_3

fib(n-1)

fib(n-2)

To compute n^{th} number

Start from 3^{rd} number

and then compute 4^{th}

then 5^{th}

⋮

till you reach n^{th}

$T(n) \in \Theta(n)$

$$T(n) = c_1 + \underline{c_2 \cdot n} + c_3$$

$c_2 \cdot (n-2)$

Amit Chintamani Awekar

+111 | VG | PK | Amit Chintamani Awekar | KH KILAPARTHI HEEMAN... | AV ANINDYA VIJAYVARG... | PS PATEL HETANG SANJ... | LA LOVISH AGGARWAL | ST

01:23:13 Leave

3:20 PM Mon 9 Aug 90%

04 Algorithm Analysis

203 course administration | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Representation | 04 Algorithm Analysis

$(1.6)^5 = 10$

n	fib S	fib R
1	1	$(1.6)^1 \rightarrow 1.6$
10	10	$(1.6)^{10} \rightarrow 10^2 \rightarrow 10^{-8}$ sec 10^{10} $\frac{\text{ins}}{\text{second}}$
		$(1.6)^{100} \rightarrow 10^{20} \rightarrow 10^{10}$ sec $\rightarrow 10 \times 10^9$
100	100	$(1.6)^{1000} \rightarrow 10^{200} \rightarrow 10^{190}$ sec
1000	1000	$(1.6)^{10000} \rightarrow 10^{2000} \rightarrow 10^{1990}$ sec
10 000	10 000	

$\downarrow$

Less than 1 sec

Amit Chintamani Awekar

+110 VG PK Amit Chintamani Awekar KH KILAPARTHI HEEMAN... AV ANINDYA VIJAYVARG... PS PATEL HETANG SANJ... LA LOVISH AGGARWAL ST

Abstractions

- Bits → Digital circuit
- Files → Secondary storage
- Machine code → Operations by Processor
- Basic data types → Data handled by processor

01:40:45

Leave

3:38 PM Mon 9 Aug

86%

05 ADT

203 course administrati... | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Represent... | 04 Algorithm Analysis | 05 ADT

- Ease of understanding
- Independence from implementation
- Separation of concerns

  specification and implementation

- Ability to build more abstraction

Amit Chintamani Awekar

+111 KH ML PS

Amit Chintamani Awekar

GUNJAN DHANUKA

SR SHUBHRADEEP ROY

PK PRADEEP KUMAR

SIDDHARTH GAUR

ST

01:49:09

Leave

3:47 PM Mon 9 Aug

05 ADT

203 course administrati... | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Represent... | 04 Algorithm Analysis | 05 ADT

Abstract Data Type (ADT)

- Data
- Operations

int   Integer

Data → 16 / 32 / 64 bits → Circuits ↑ Transister ↑

Operations → +, −, ×, /, ++, ==

- Access to data is only through operations

Amit Chintamani Awekar

+111 KH ML PS

Amit Chintamani Awekar

GUNJAN DHANUKA

SHUBHRADEEP ROY

PRADEEP KUMAR

SIDDHARTH GAUR

ST

Abstract Data Type (ADT)

- Data
- Operations

int — Integer

Data → 16 / 32 / 64 bits → Circuits ↑ Transister ↑

Operations → +, -, x, /, ++, ==

- Access to data is only through operations
- Implementation details need not be shared
- Naturally translates to OOP languages

02:00:13

Leave

3:58 PM Mon 9 Aug

05 ADT

203 course administrati... | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Represent... | 04 Algorithm Analysis | 05 ADT

Array ADT

Data: < index, value >

Collection of pairs

Operations:

createNew (capacity, type)

StoreValue (index, val)

accessValue (index)

searchValue (value)

initArray (value)

- Store same type of data
- Contiguous storage
- fixed storage
- Access element with index
- Change element at any index
- Structure is fixed at the time of creation
- We can have array of arrays
- Most of the programming languages provide it

Amit Chintamani Awekar

+109 AK M Amit Chintamani Awekar PK PRANSHU KANDOI TAPAN SETHI PP PRAKHAR PANDEY PRANAV NAIR ST

02:00:45

Leave

3:58 PM Mon 9 Aug

05 ADT

203 course administrati... | syllabus | 02 Model Of Computation | 008 example | 03 Algorithm Represent... | 04 Algorithm Analysis | 05 ADT

Array ADT

Data: <index, value>

Collection of pairs

Operations:

createNew (capacity, type)

StoreValue (index, val)

accessValue (index) 5 → 2.9

searchValue (value)

initArray (value)

- Store same type of data
- Contiguous storage
- fixed storage
- Access element with index
- Change element at any index
- Structure is fixed at the time of creation
- We can have array of arrays
- Most of the programming languages provide it

Amit Chintamani Awekar

+109 AK M Amit Chintamani Awekar PK PRANSHU KANDOI TAPAN SETHI PP PRAKHAR PANDEY PRANAV NAIR ST

02:09:57

05 ADT

Array ADT

Data: < index, value >

Collection of pairs

Operations:

createNew (capacity, type)

StoreValue (index , val)

accessValue (index)

searchValue (value)

initArray (value)

- ✓ · Store same type of data
- ✗ · Contiguous Storage
- ✗ · fixed Storage
- ✓ · Access element with index
- ✓ · Change element at any index
- ✗ · Structure is fixed at the time of creation
- ✓ · We can have array of arrays
- · Most of the programming languages provide it

Amit Chintamani Awekar

+107 AK M PK Amit Chintamani Awekar TAPAN SETHI PRAKHAR PANDEY PRANAV NAIR PANKAJ YADAV ST