

COL702: Advanced Data Structures and Algorithms

(Semester-I-2024-25)

Grading

Grading component	#	%
Homework	5	20% (best n-1 out of n)
Quiz	5	20% (best n-1 out of n)
Minor	1	20%
Final	1	35%
Comprehension quiz	See below	3%
Attendance	-	2%

- *Comprehension quizzes are online quizzes conducted through **Moodle/Gradescope** that students are supposed to take for a better understanding of the material.*

Logistics

Grading component	Comments
Homework	<ul style="list-style-type: none">• Can be done in groups of size at most 2.• Submissions to be made over Gradescope. Please avoid missing deadlines.• We will prefer if you submit typed solutions. Latex source files will be provided.• May include a programming component.
Quiz	<ul style="list-style-type: none">• Conducted using online Gradescope feature or in class.• We may create multiple versions of the quiz.
Comprehension quiz	<ul style="list-style-type: none">• We prefer if you take the quiz soon after the lecture component has been completed. However, we will have a flexible deadline.

Important points

- Please register on Piazza (no code required).
- You will be registered on Gradescope by syncing with Moodle.
- Course webpage:
 - <https://www.cse.iitd.ac.in/~rjaiswal/Teaching/2024/COL702/>
- Textbook
 - Algorithms by *Dasgupta, Papadimitriou, and Vazirani*.
 - Algorithm Design by *Jon Kleinberg and Eva Tardos*.
 - Algorithms by *Russell Impagliazzo and Ragesh Jaiswal*
- Audit:
 - You will need to **get (C) or better grade** for Audit-pass. You should plan to take all the grading components even if you audit.

Analyzing algorithms

A royal mathematical challenge (1202):





















Suppose that rabbits take exactly one month to become fertile, after which they produce one child per month, forever. Starting with one rabbit, how many are there after n months?



Leonardo da Pisa, aka
Fibonacci

The proliferation of rabbits

Rabbits take one month to become fertile, after which they produce one child per month, forever.

	Fertile	Not fertile
Initially		
One month		
Two months		
Three months	 	
Four months	  	 
Five months	    	  

The Fibonacci sequence

$$F_1 = 1, \quad F_2 = 1, \quad F_n = F_{n-1} + F_{n-2}$$

These numbers grow *very* fast: $F_{30} > 10^6$!

In fact, $F_n \approx 2^{0.694n} \approx 1.6^n$, **exponential growth**.

The Fibonacci sequence

$$F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2}$$

Can you see why $F_n < 2^n$?

Computing Fibonacci numbers

```
function Fib1(n)
if n = 1 return 1
if n = 2 return 1
return Fib1(n-1) + Fib1(n-2)
```

A recursive algorithm

Two questions we always ask about algorithms:

Does it work correctly?

How long does it take?

Running time analysis

```
function Fib1(n)
if n = 1 return 1
if n = 2 return 1
return Fib1(n-1) + Fib1(n-2)
```

Exponential time... how bad is this?

Eg. Computing F_{200} needs about 2^{140} operations.
How long does this take on a fast computer?

IBM Summit



Can perform up to 200 quadrillion ($= 200 \times 10^{15}$) operations per second.

Is exponential time all that bad?

The Summit needs 2^{82} seconds for F_{200} .

Time in seconds

2^{10}

2^{20}

2^{30}

2^{40}

2^{45}

2^{51}

2^{57}

2^{60}

Interpretation

17 minutes

12 days

32 years

cave paintings

homo erectus discovers fire

extinction of dinosaurs

creation of Earth

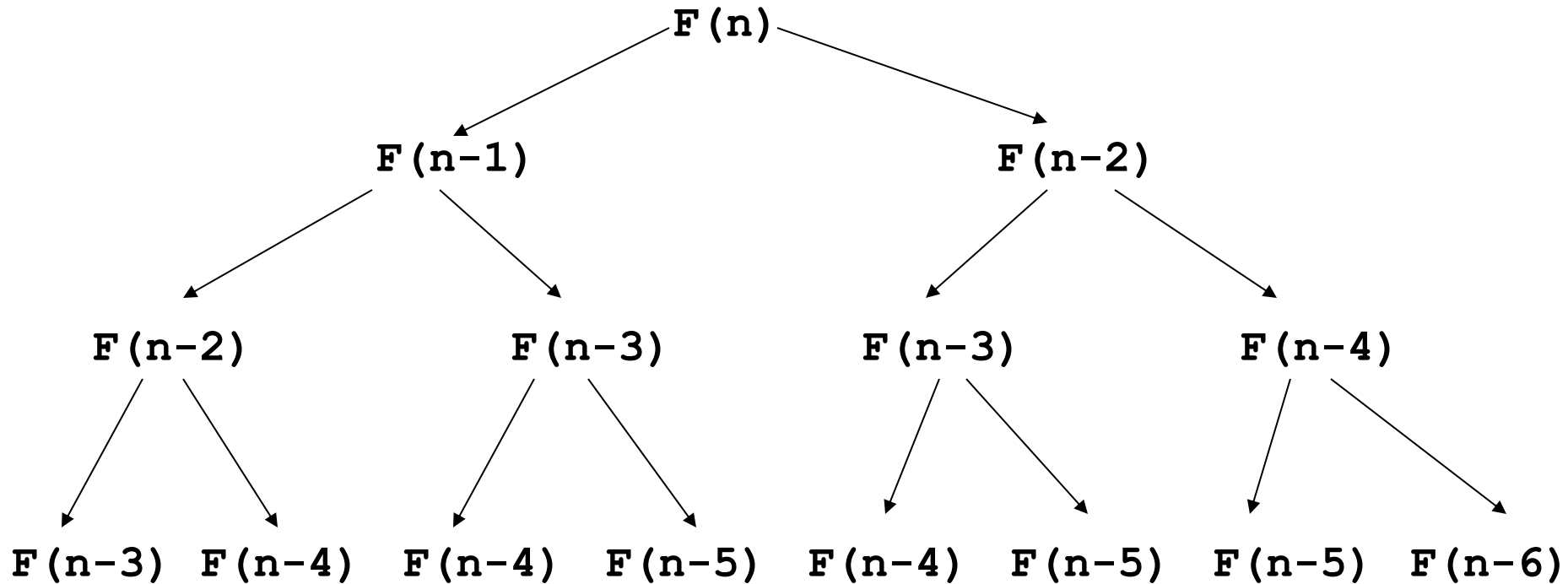
origin of universe

Post mortem

What takes so long?

Let's unravel the recursion...

```
function Fib1(n)
  if n = 1 return 1
  if n = 2 return 1
  return Fib1(n-1) + Fib1(n-2)
```



The same subproblems get solved over and over again!

A better algorithm

Subproblems: F_1, F_2, \dots, F_n . Solve them in order and *save their values!*

```
function Fib2(n)
  Create an array fib[1..n]
  fib[1] = 1
  fib[2] = 1
  for i = 3 to n:
    fib[i] = fib[i-1] + fib[i-2]
  return fib[n]
```

[1] Does it return the correct answer?

[2] How fast is it?

Polynomial vs. exponential

Polynomial running times:

Exponential running times:

To an excellent first approximation:

polynomial is reasonable

exponential is not reasonable

This is one of the most fundamental dichotomies in the analysis of algorithms.