



BITS F232: FOUNDATIONS OF DATA STRUCTURES & ALGORITHMS (1ST SEMESTER 2023-24) INTRODUCTION

Chittaranjan Hota, PhD
Sr. Professor of Computer Sc.
BITS-Pilani Hyderabad Campus
[hota\[AT\]hyderabad.bits-pilani.ac.in](mailto:hota[AT]hyderabad.bits-pilani.ac.in)

WHY SHOULD YOU STUDY THE COURSE?



Image source: <https://www.quora.com/>



You will be able to analyze a program to find out where to improve.

Searching your
name in the
MCN list.

Driving through Hyd using maps.

FIBONACCI: RECURSIVE OR ITERATIVE?

Algorithm 2: $F(n)$

Input: Some non-negative integer n

Output: The n th number in the Fibonacci Sequence

$A[0] \leftarrow 0;$


$A[1] \leftarrow 1;$

for $i \leftarrow 2$ **to** $n - 1$ **do**

$A[i] \leftarrow A[i - 1] + A[i - 2];$

return $A[n - 1]$







WHAT KIND OF PROBLEMS CAN YOU SOLVE?



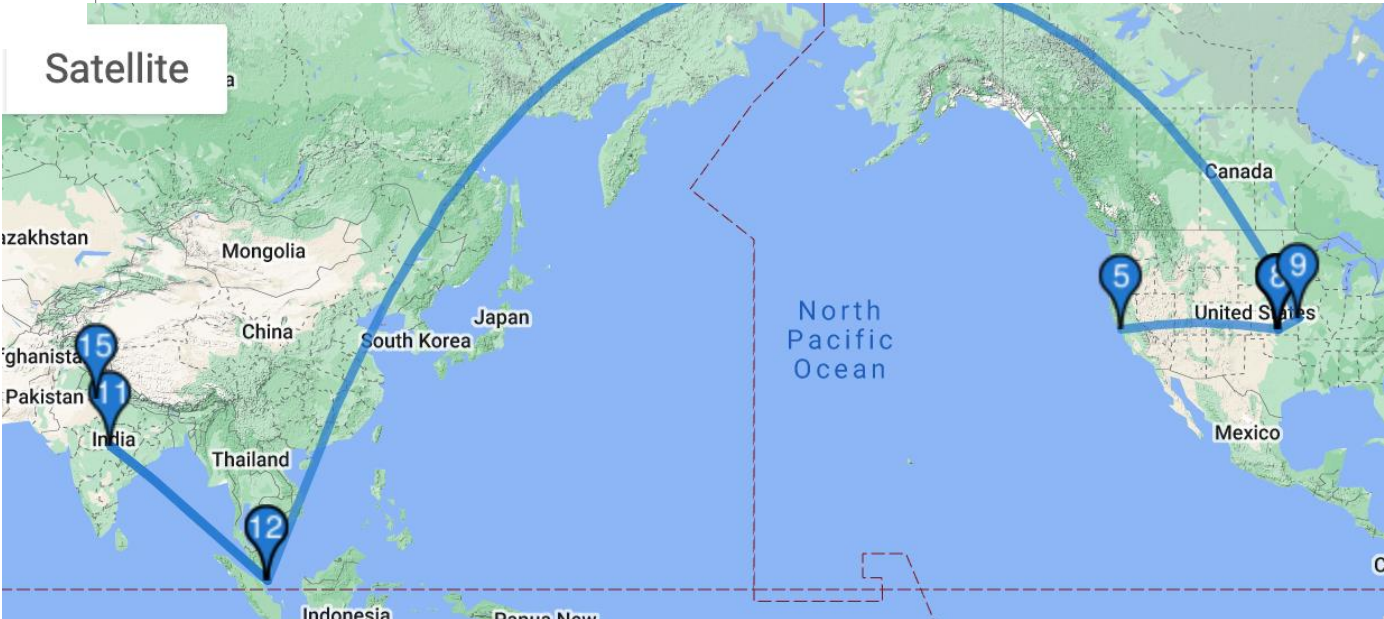
Rice

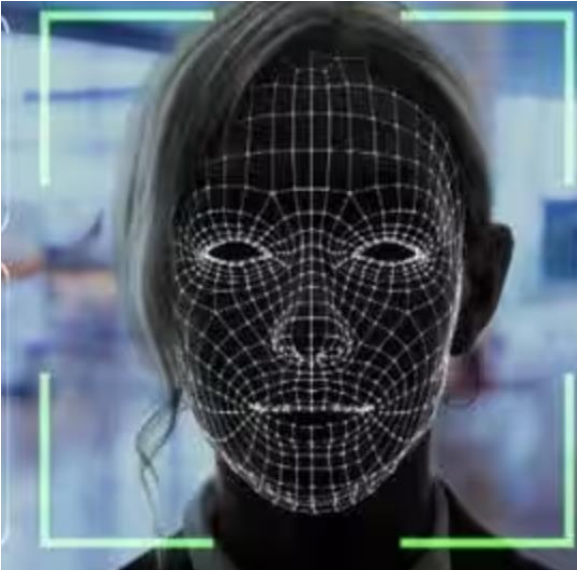
Showing results for *Rice*

some more suggestions


	SUPER SAVER Dosa - Rice	5 kg	Rs.258		Qty 1	ADD 
	SUPER SAVER Dosa - Rice	1 kg	Rs.53		Qty 1	ADD 

Satellite



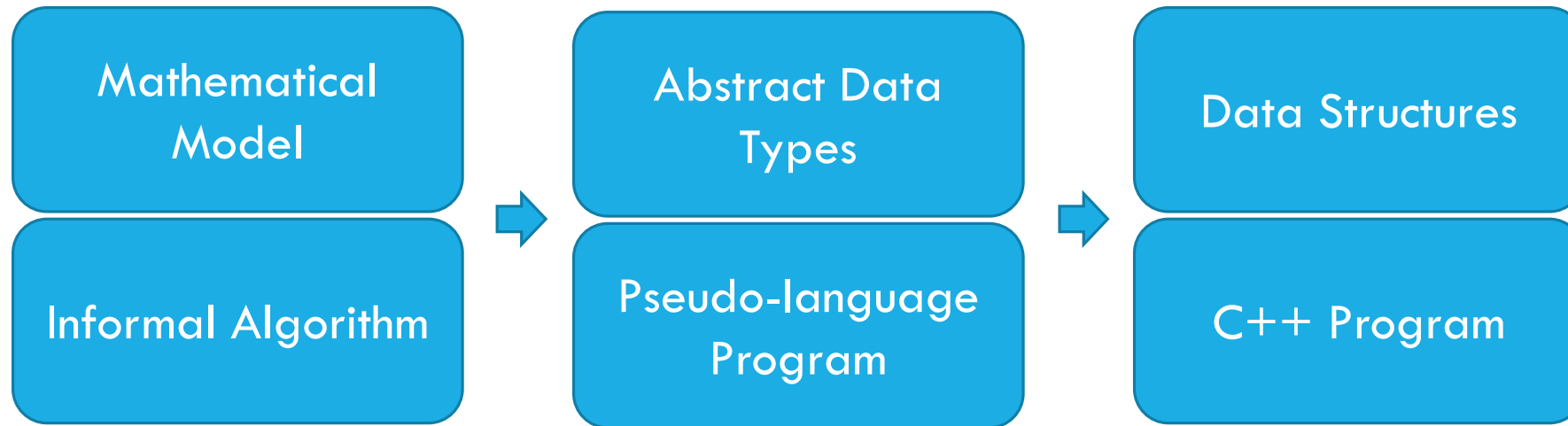


DigiYatra

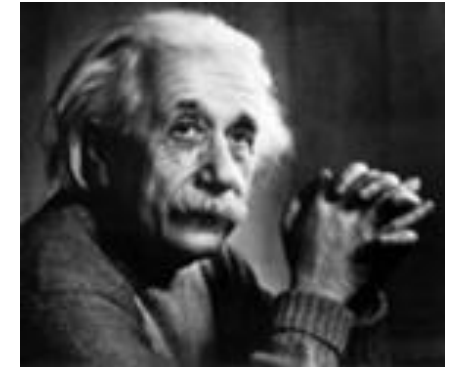


<https://gsuite.tools/traceroute>

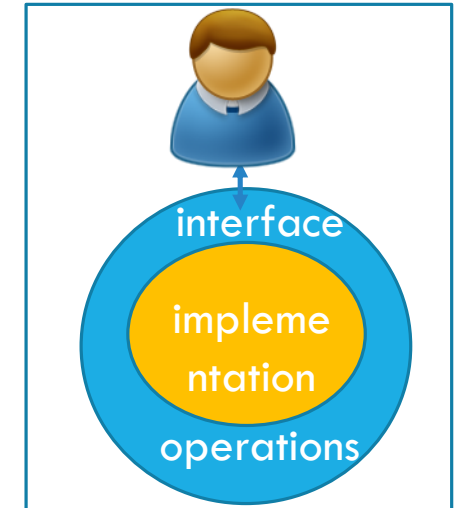
THE PROBLEM SOLVING PROCESS



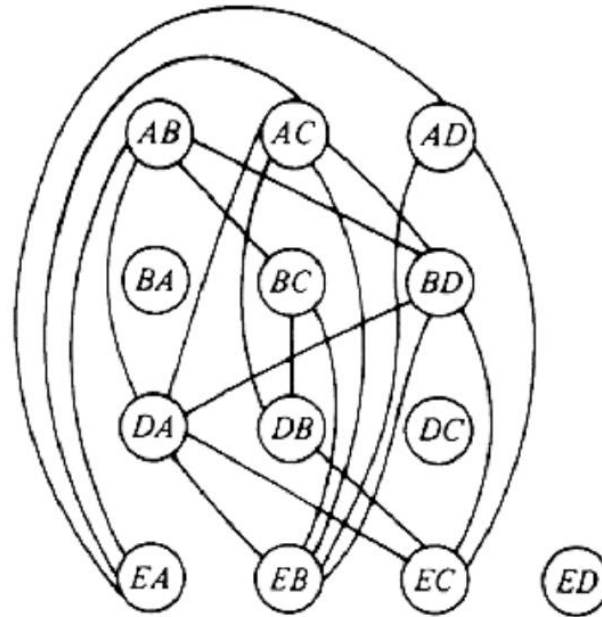
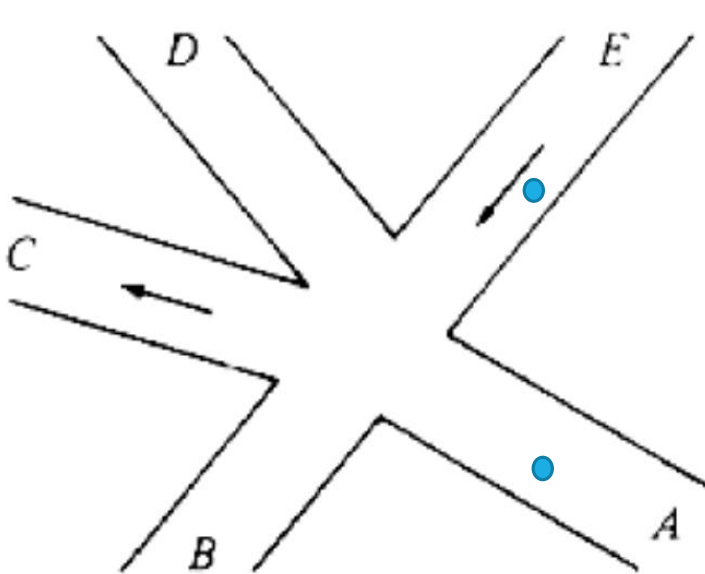
- finding currents in electrical circuits, predicting spread of covid-19
- Language translator
- An abstract data type (**ADT**), is a logical description of how we view the data and the operations that are allowed without knowing how they will be implemented.
- Data structure (physical description) is the implementation of ADTs.



If I had 1 hour to save the world, ...



AN EXAMPLE: A TRAFFIC LIGHTING SYSTEM



1. Select an uncolored vertex and color it with a new color.
2. Scan the list of uncolored vertices. For each uncolored vertex, determine whether it has an edge to any vertex already colored with the new color. If there is no such edge, color the present vertex with the new color.

(Problem of road intersection) (Graph with incompatible turns) (Greedy coloring algorithm)

- The approach is called "greedy" because it colors a vertex whenever it can, without considering the potential drawbacks inherent in making such a move.
AB,AC,AD,BA,DC,ED;

CONTINUED...

```
SET greedy_graph_coloring (Input:G:GRAPH,  
Output: Newclr: SET) {  
    Newclr  $\leftarrow \emptyset$ ;  
    for (each uncolored vertex 'v'  $\in$  G)  
    {  
        if 'v' is not adjacent to any vertex in Newclr  
        {  
            v  $\leftarrow$  colored;  
            Newclr  $\leftarrow$  Newclr  $\cup$  'v'  
        }  
    }  
}
```

(Pseudo Code)

<https://www.techiedelight.com/greedy-coloring-graph/>

GRAPH **ADT**: G

1. graphNew(): creating a graph
2. addVertex (v)
3. addEdge (v1, v2)
4. getVertex (unclored)
5. markVertex (colored)
6. ...

	AB	AC	AD	BA	BC	BD	DA	DB	DC	EA	EB	EC	ED
AB					1	1	1			1			
AC						1	1	1		1	1		
AD										1	1	1	
BA													
BC	1										1		
BD	1	1									1	1	
DA	1	1				1					1	1	
DB		1			1							1	
DC													
EA	1	1	1										
EB		1	1		1	1	1						
EC			1			1	1	1					
ED													

(Adjacency matrix to implement the graph)



THANK YOU!

Next Class: Introduction to C++