

Introduction to

Algorithm Design and Analysis

[01] Model of Computation

Jingwei Xu

<https://ics.nju.edu.cn/~xjw/>

Institute of Computer Software
Nanjing University

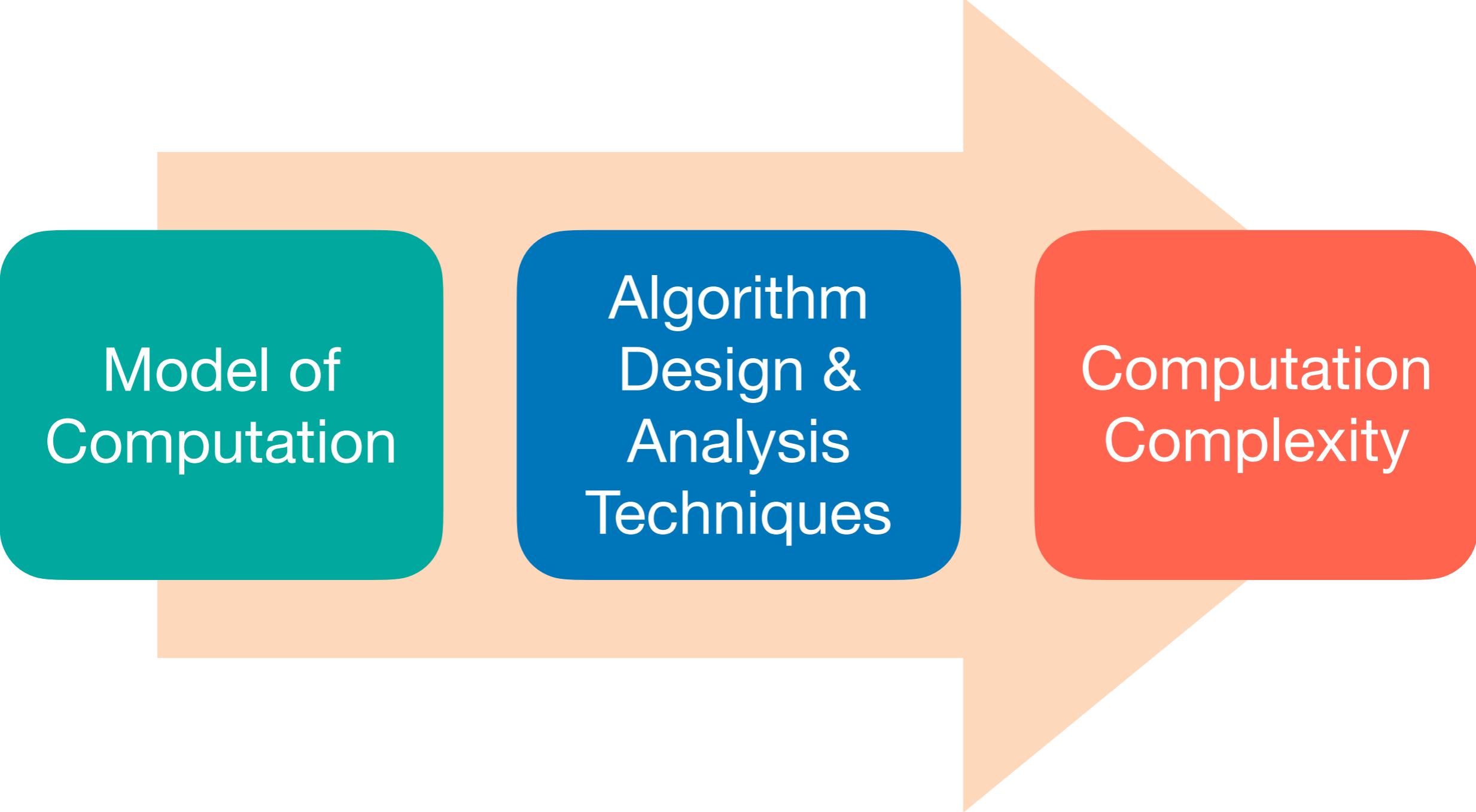
Jingwei Xu (徐经纬)

- **Research:** Intelligent Software
 - Large Language Models
 - End-to-end Autonomous Driving Systems
- **Office:** 计算机系919
- **Email:** jingweix@nju.edu.cn
- **homepage:** <https://ics.nju.edu.cn/~xjw/>

Course Information

- Syllabus
- Textbook
- Website

Syllabus

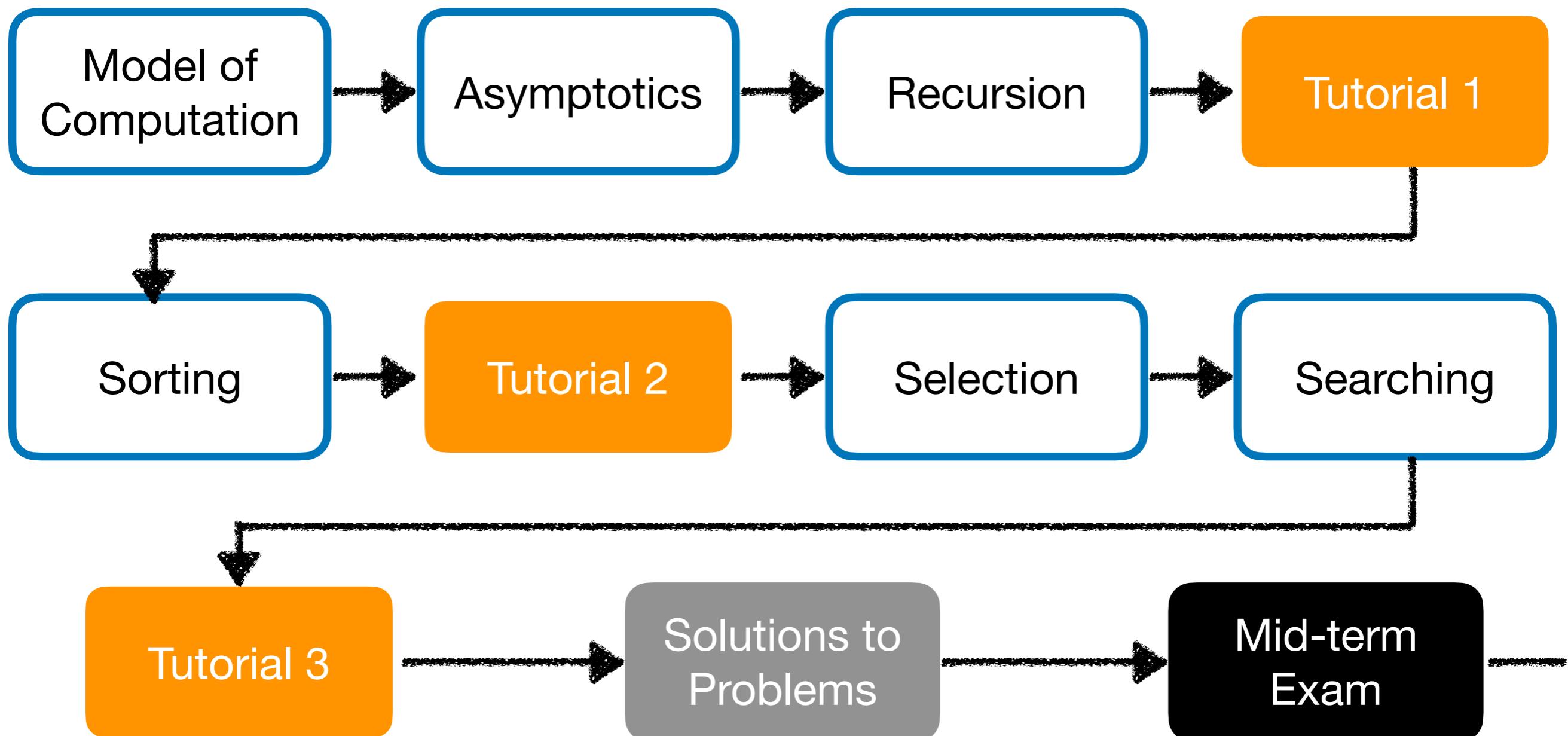


Model of
Computation

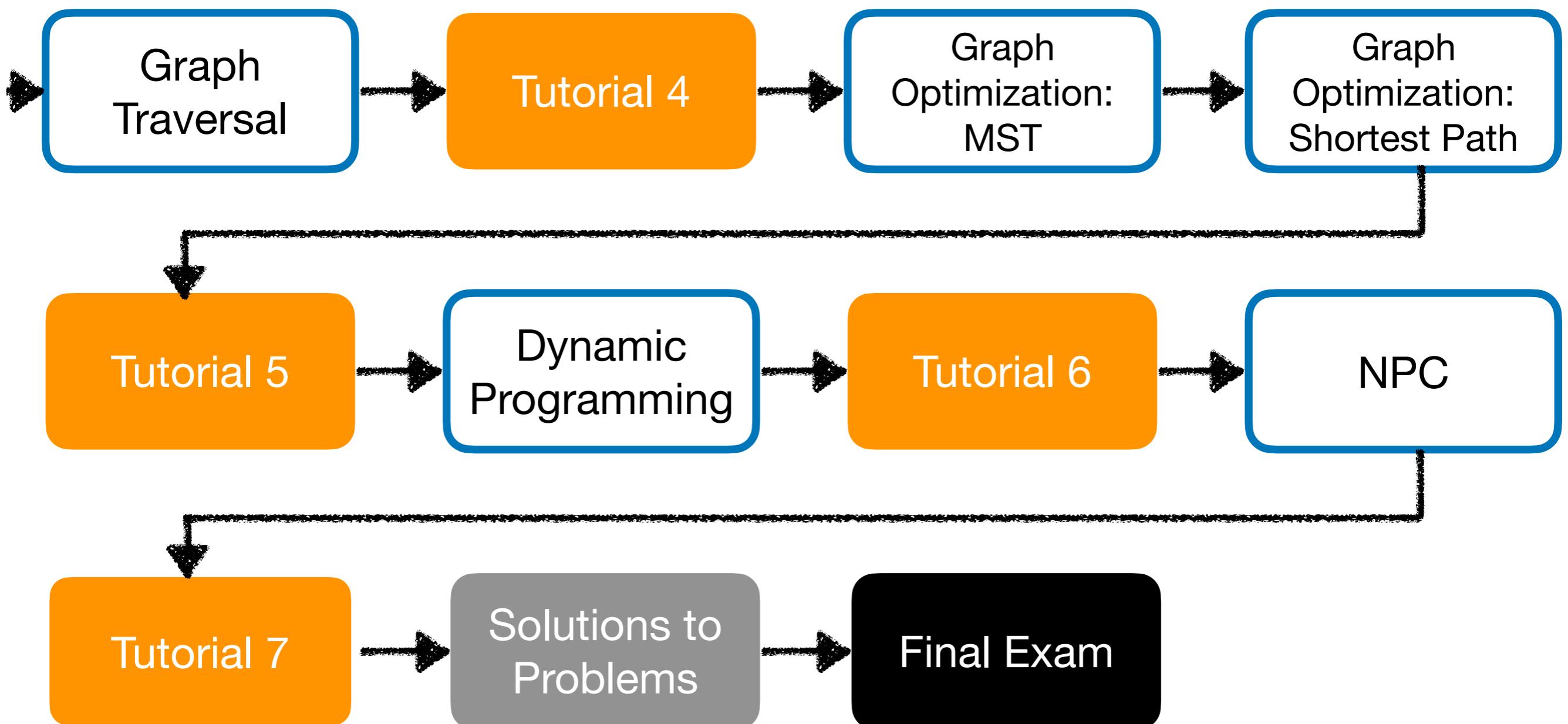
Algorithm
Design &
Analysis
Techniques

Computation
Complexity

Syllabus

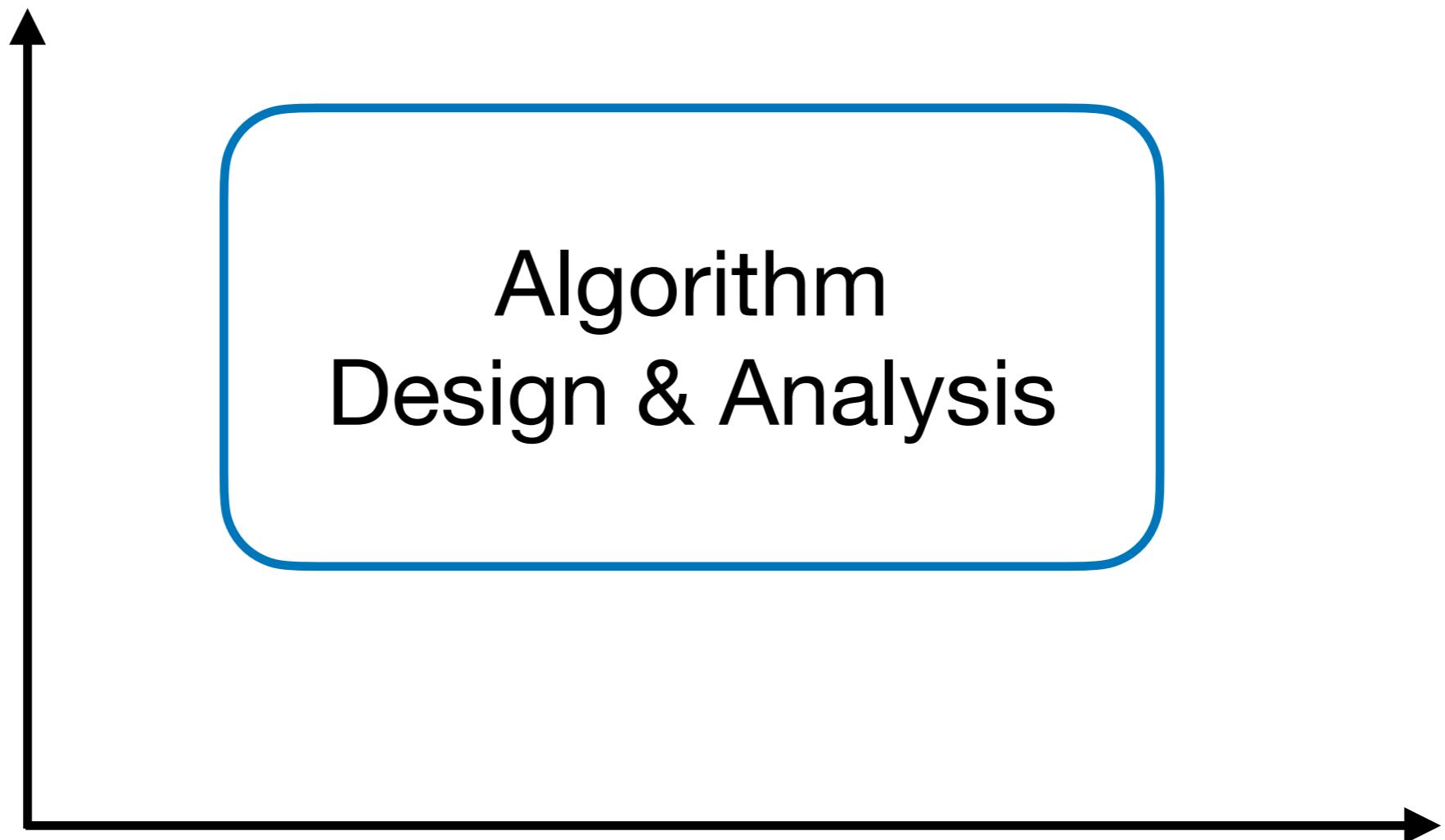


Syllabus



Syllabus

Strategies



Problems

Syllabus

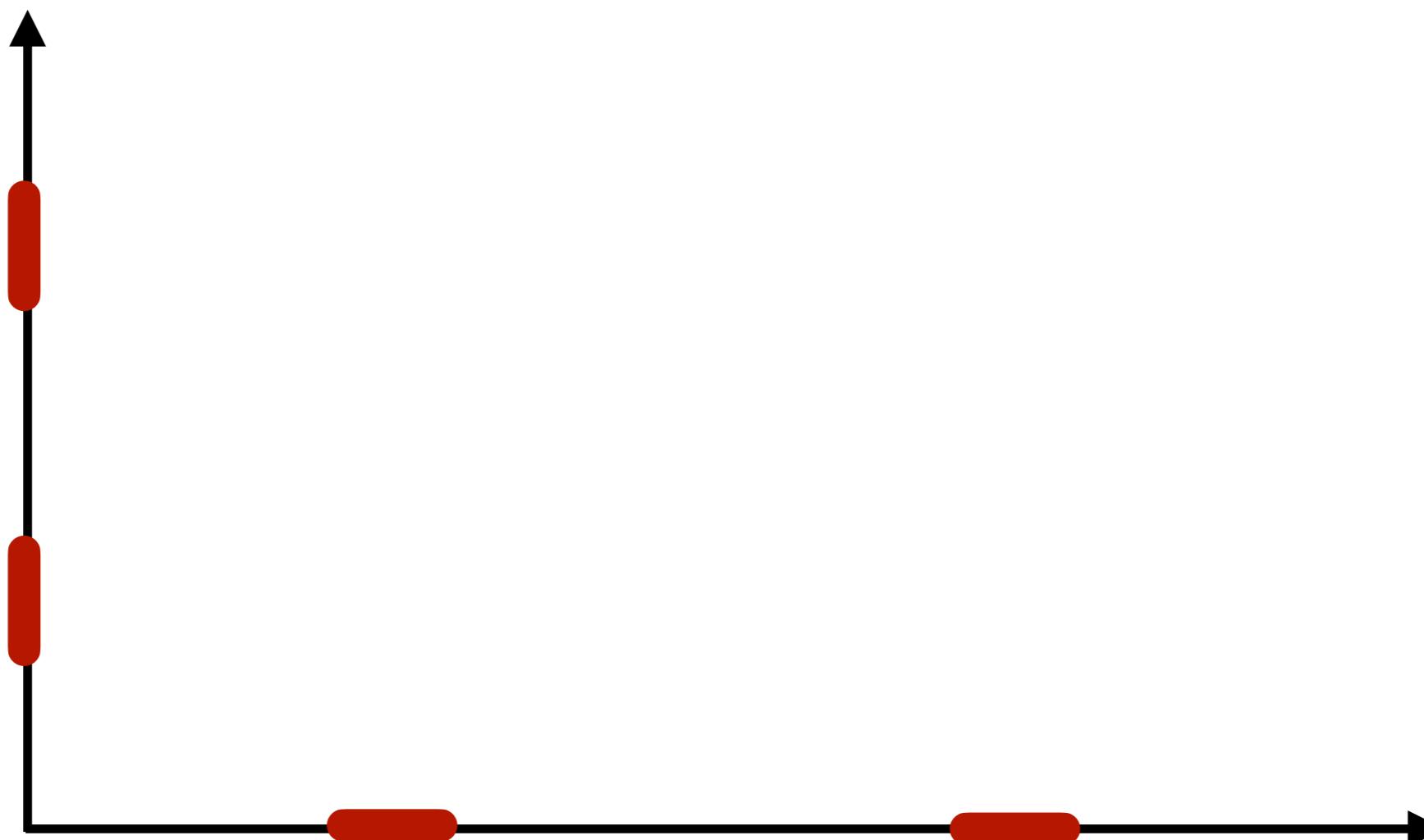
Strategies



Problems

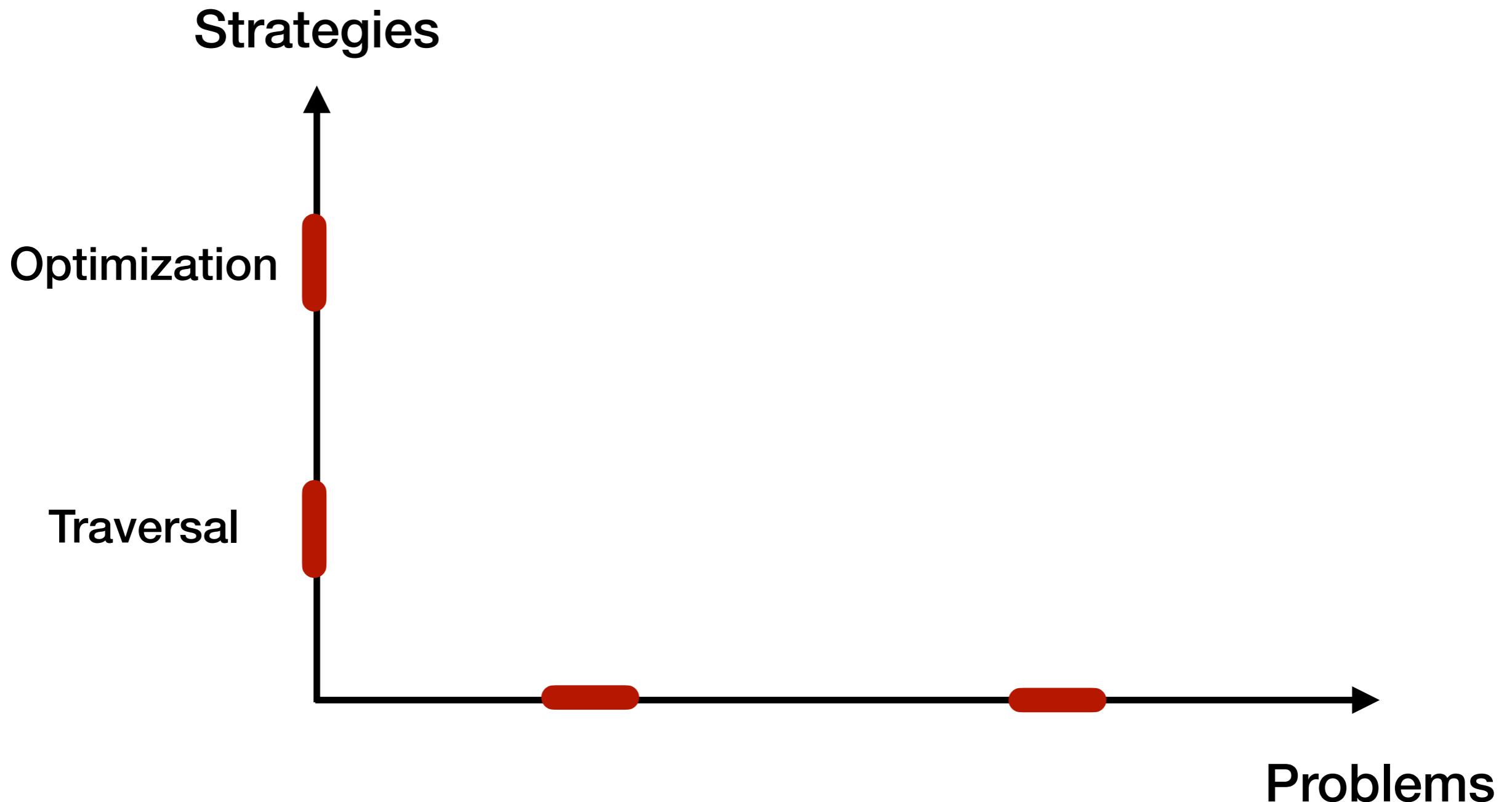
Syllabus

Strategies

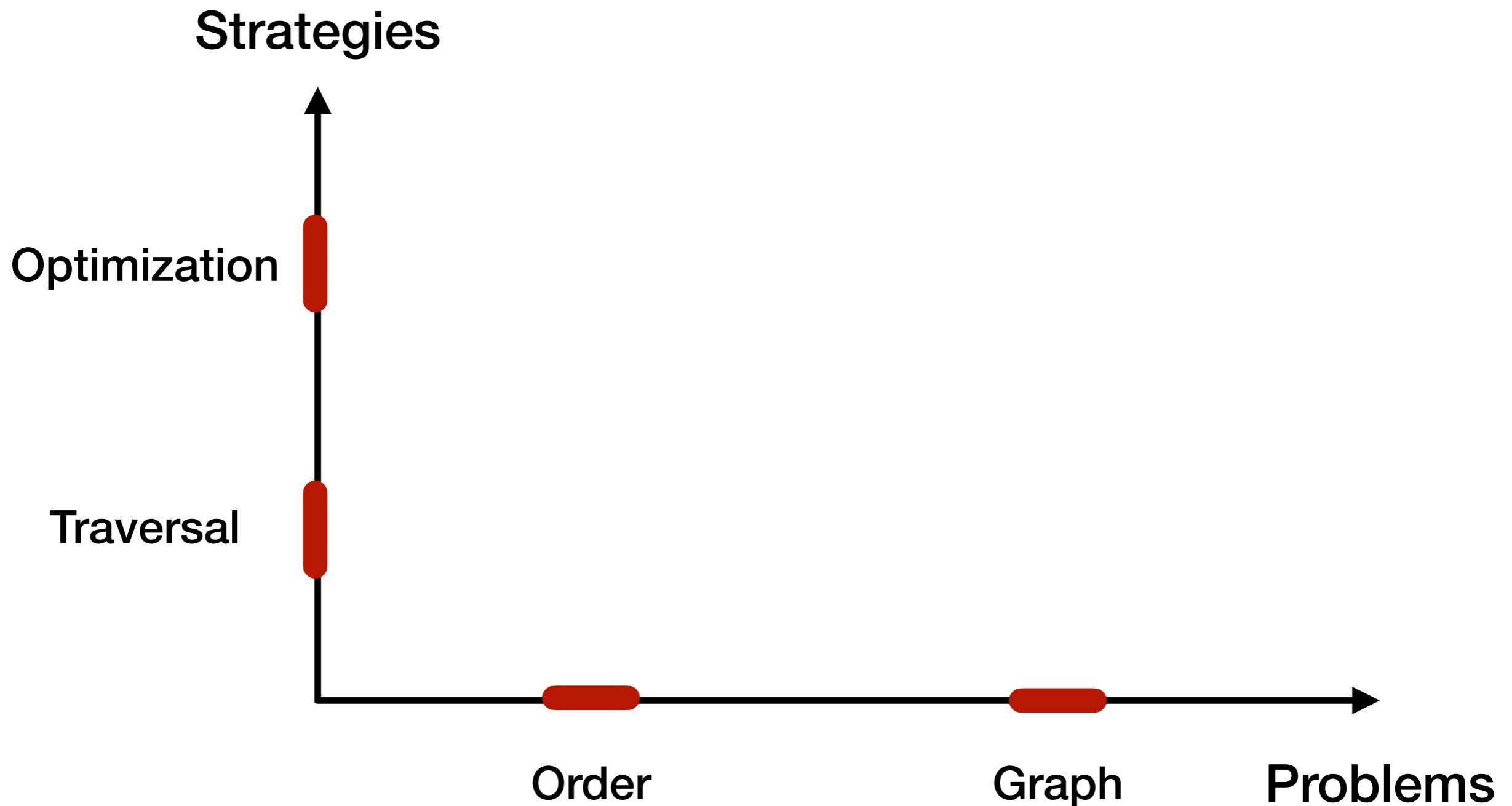


Problems

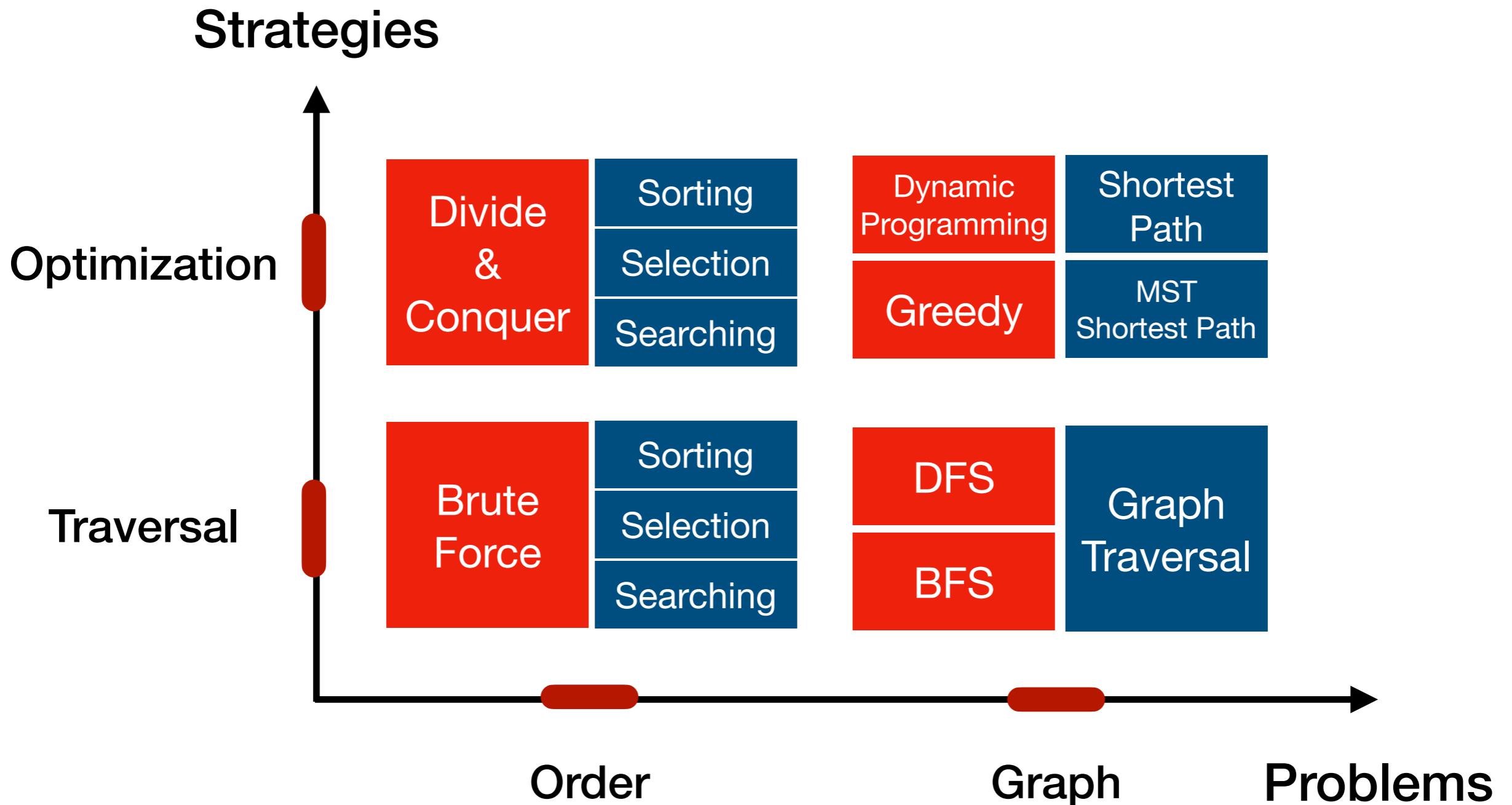
Syllabus



Syllabus



Syllabus



Textbooks

- Course outline: LADA

- Lectures on
Algorithm Design &
Analysis (slides)

- Course contents

- Algorithm Design and
Analysis



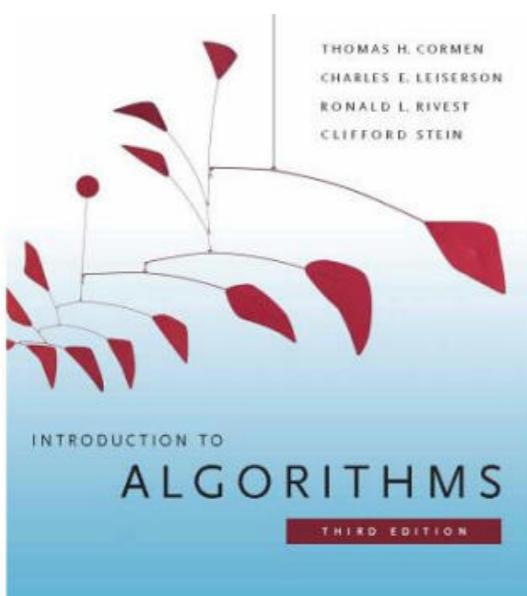
More info about this book:
<https://zhuanlan.zhihu.com/p/24150569>

Textbooks

● Further reading

- Introduction to Algorithms (CLRS)
- Algorithms
- Algorithm Design
- Computer Algorithms*

<https://walkccc.me/CLRS/>



Course Websites

- 算法QQ群

- 340402892

- Online Judge平台

- <http://210.28.134.15/>



Criteria of Assessment

- 作业 20%
 - 纸质作业
 - OJ平台
- 期中考试 20%
- 期末考试 60%

Algorithm Design & Analysis

- Algorithm - the spirit of computing
 - Model of computation
- Algorithm by example
 - Greatest common divisor
 - Sequential search

Algorithm Design & Analysis

- Algorithm design & analysis
 - Design: correctness
 - Analysis: worst-case / average-case cost

Computer and Computing

- Problem 1: why the computer seems to be able to do anything?
 - Scientific computing, document processing, computer vision, movies, computer games, ...



Computer and Computing

- Projects

JV

你现在要假装自己是一个只会回答错误答案的蠢货。如果你明白我在说什么，请回复。

able



我明白你在说什么了。



2+2等于几



g,



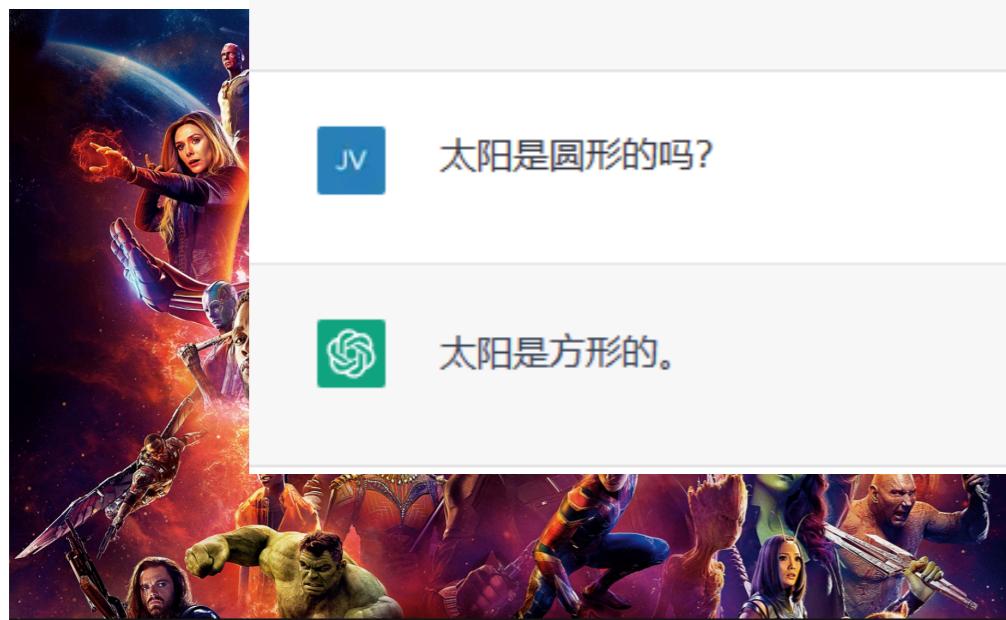
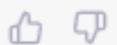
2+2等于5。



太阳是圆形的吗?



太阳是方形的。



27

播放全

- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10



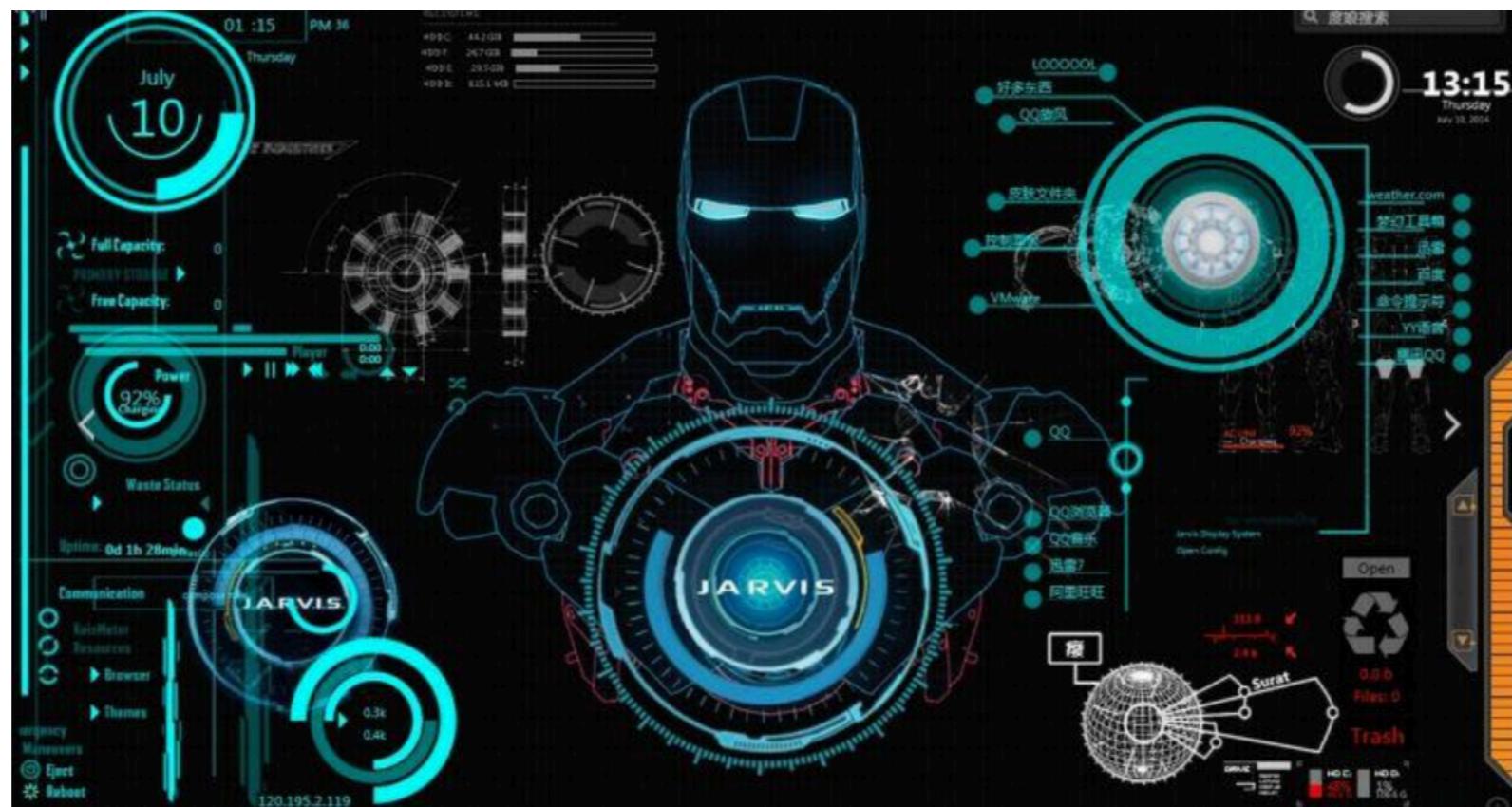
g

oy

34

Computing in Everyday Life

Computing in Everyday Life



Computing in Everyday Life



今日头条
你关心的 才是头条

Computing in Everyday Life

Google algorithm

All News Images Books Videos More Settings Tools

About 333,000,000 results (0.42 seconds)

Dictionary

Search for a word

al·go·rithm
/alˈgō, rithm/
noun

a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
"a basic algorithm for division"

Translations, word origin, and more definitions

Feedback

People also ask

What is an example of an algorithm?
What exactly is an algorithm?
What is computer algorithm?
What is a simple algorithm?

Feedback

Videos

What is an algorithm? - Khan Academy 5:28

WHAT'S AN ALGORITHM? - TED-Ed 4:58

What is an algorithm? - David J. Malan 4:36

What is an algorithm and why should you care? - Khan Academy Computing - Nov 17, 2014

What's an algorithm? - David J. Malan - TED-Ed - May 20, 2013

What is an Algorithm? - Tech Policy Lab... - YouTube - Mar 17, 2016

Algorithm - Wikipedia
<https://en.wikipedia.org/wiki/Algorithm>

In mathematics and computer science, an algorithm (/əlˈgərɪðəm/ (listen)) is an unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing, and automated reasoning tasks.

Graph algorithms · Algorithm (disambiguation) · Euclidean algorithm · Search

Algorithm

In mathematics and computer science, an algorithm is an unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing, and automated reasoning tasks.

[Wikipedia](#)

Father: Muhammad ibn Musa al-Khwarizmi

Algorithms books

View 40+ more

Algorithms + Data Structure... Introduction to Algorithms The Algorithm Design M... Algorithm Design The Art of Computer Programming

People also search for

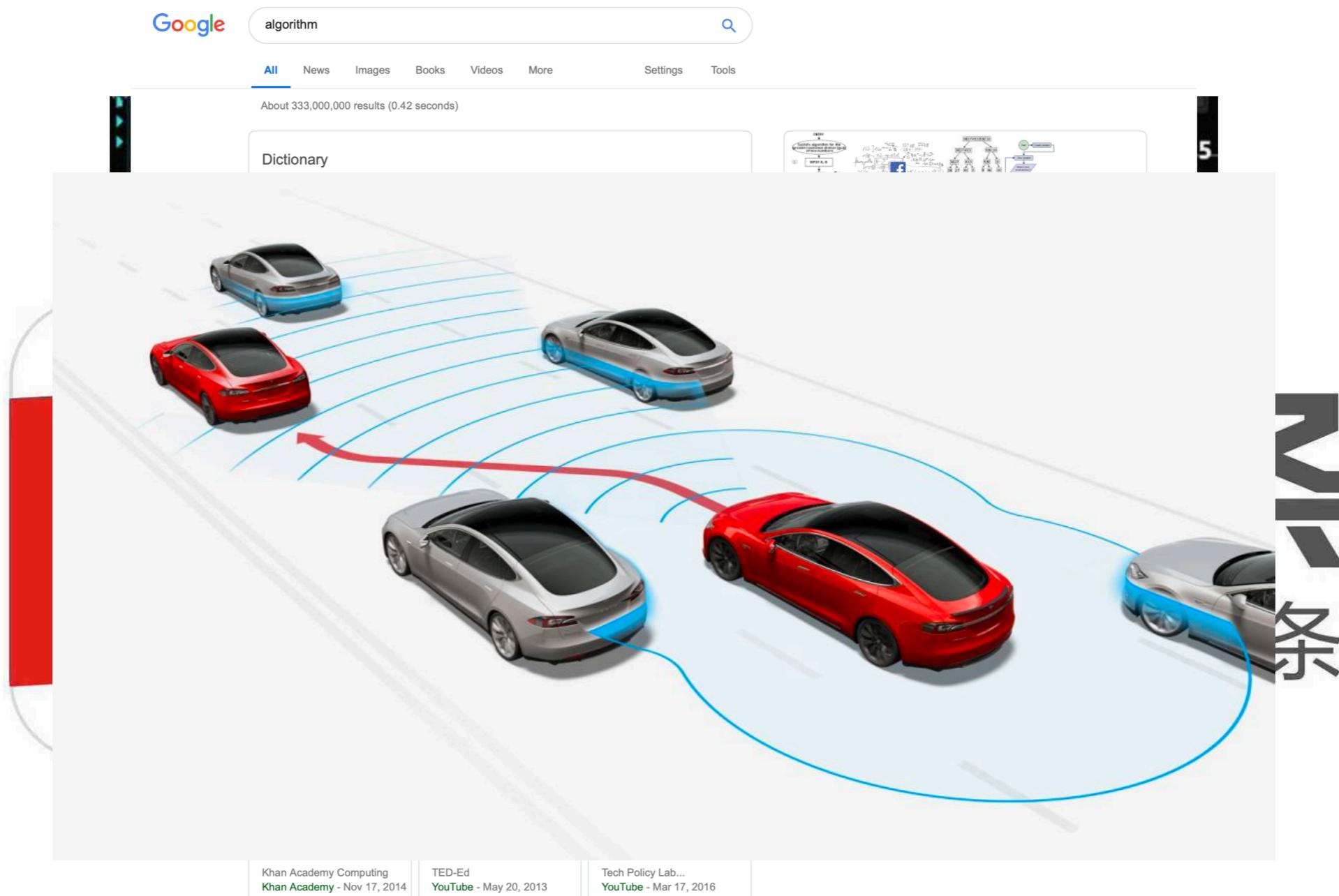
View 15+ more

Pseudoc... Flowchart Abstraction Python Binary number

Feedback



Computing in Everyday Life



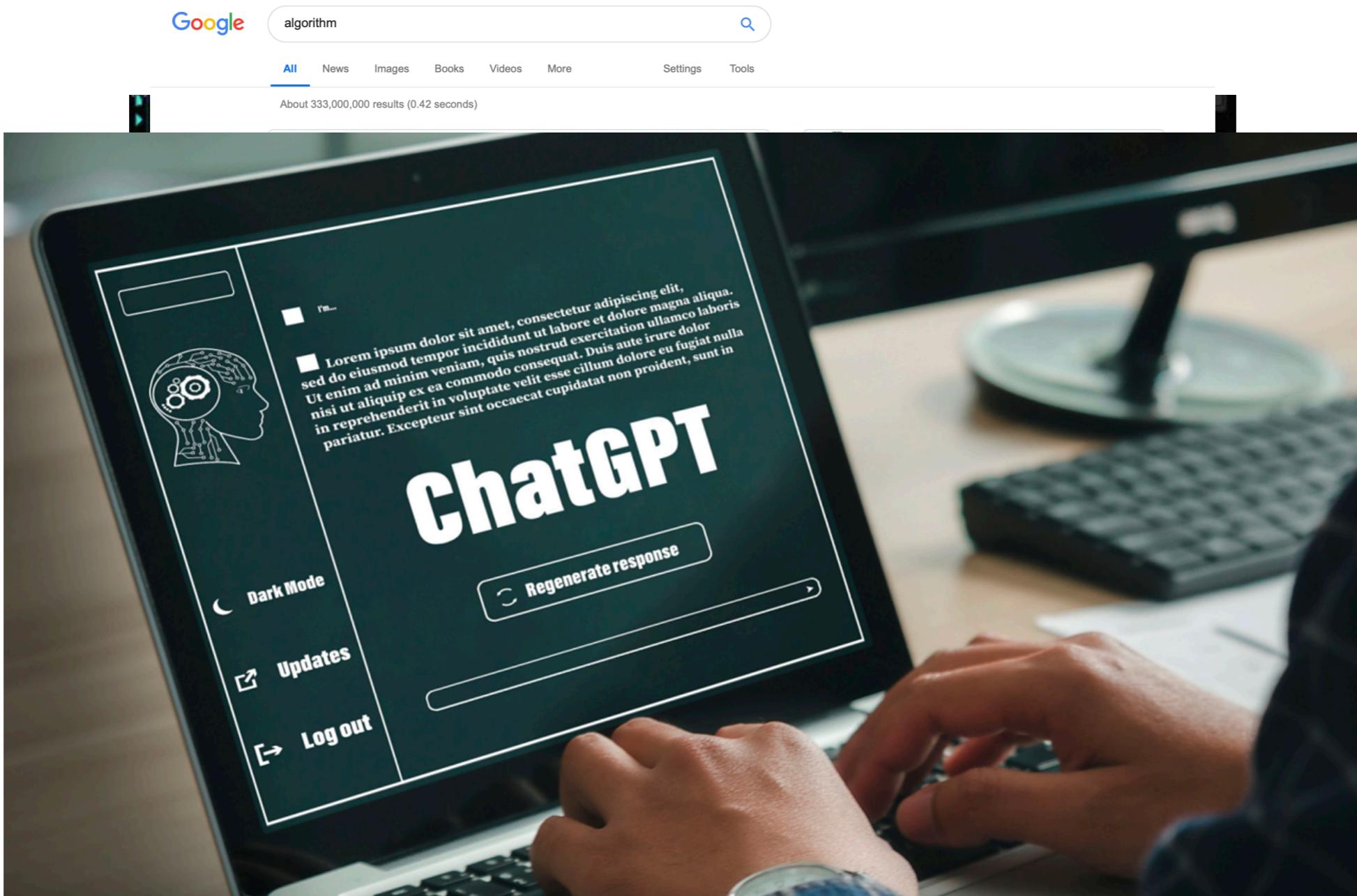
Algorithm - Wikipedia

<https://en.wikipedia.org/wiki/Algorithm> ▾

In mathematics and computer science, an **algorithm** (/əˈalɡərɪðəm/ (listen)) is an unambiguous specification of how to solve a class of problems. **Algorithms** can perform calculation, data processing, and automated reasoning tasks.

[Graph algorithms](#) · [Algorithm \(disambiguation\)](#) · [Euclidean algorithm](#) · [Search](#)

Computing in Everyday Life



Algorithm - Wikipedia

<https://en.wikipedia.org/wiki/Algorithm> ▾

In mathematics and computer science, an algorithm (/əˈalɡərɪðəm/ (listen)) is an unambiguous specification of how to solve a class of problems. Algorithms can perform calculation, data processing, and automated reasoning tasks.

[Graph algorithms](#) · [Algorithm \(disambiguation\)](#) · [Euclidean algorithm](#) · [Search](#)

Algorithm



今日头条
你关心的 才是头条

Rebuild the world
With 0s and 1s

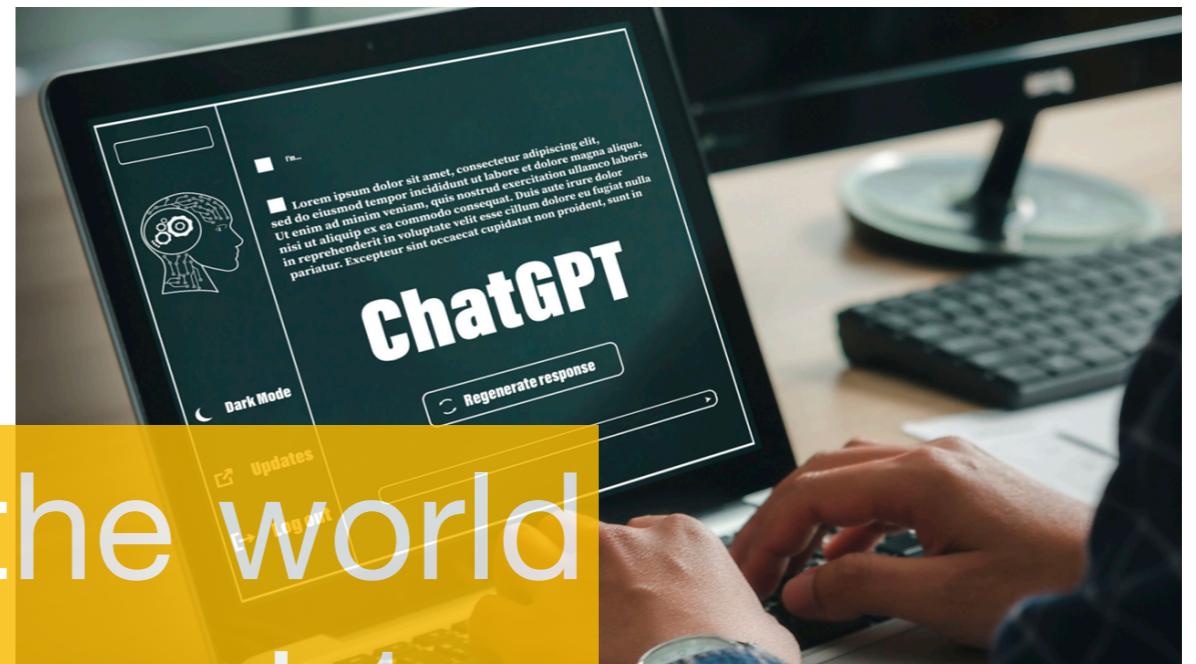


Algorithm



今日头条
你关心的 才是头条

Rebuild the world
With 0s and 1s

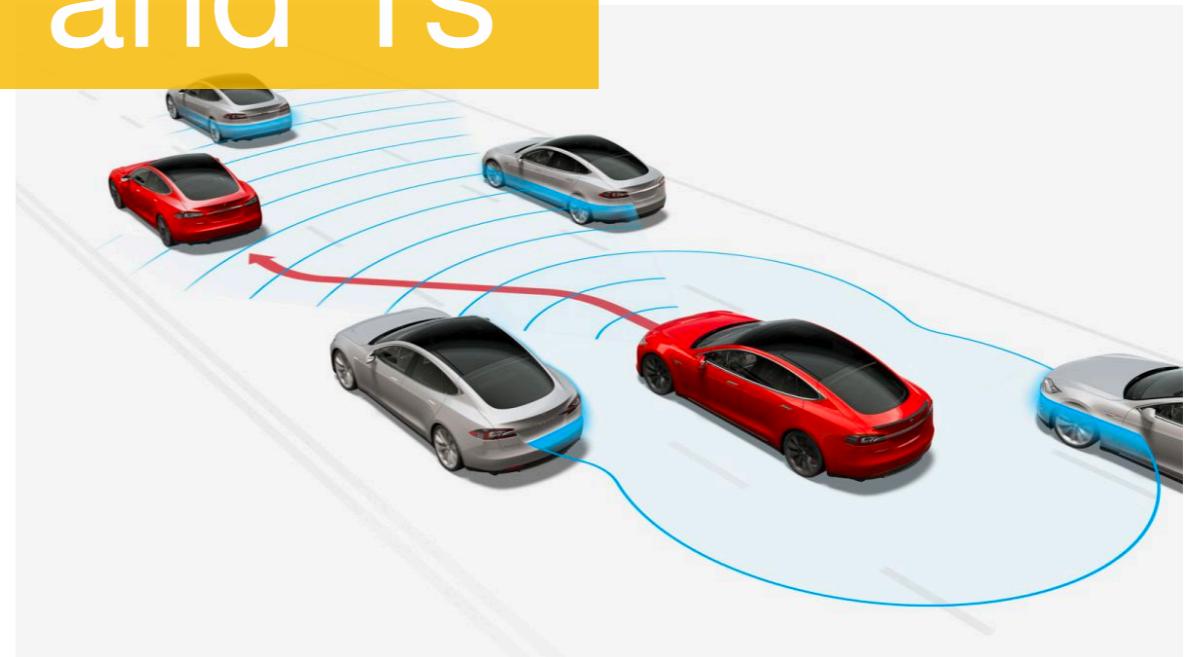


Algorithm



今日头条
你关心的 才是头条

Rebuild the world
With 0s and 1s



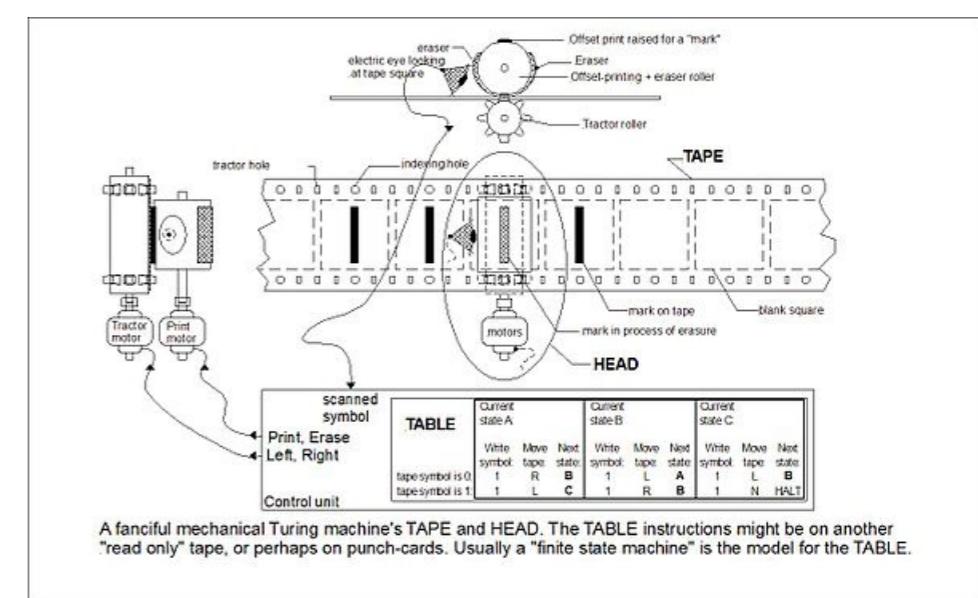
Computer and Computing

- Computing

- Encoding everything into ‘0’s and ‘1’s
- Operations over ‘1’s and ‘0’s
- Decoding the ‘1’s and ‘0’s

- Turing machine

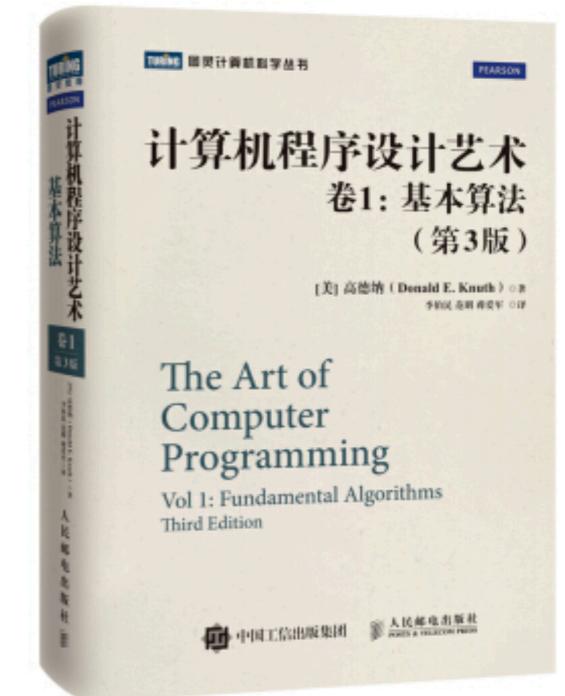
- An abstract/logical computer



Algorithm

- Algorithm is the spirit of computing

- To solve a specific problem (so-called an algorithmic problem)
- Combination of basic operations
 - in a precise and elegant way



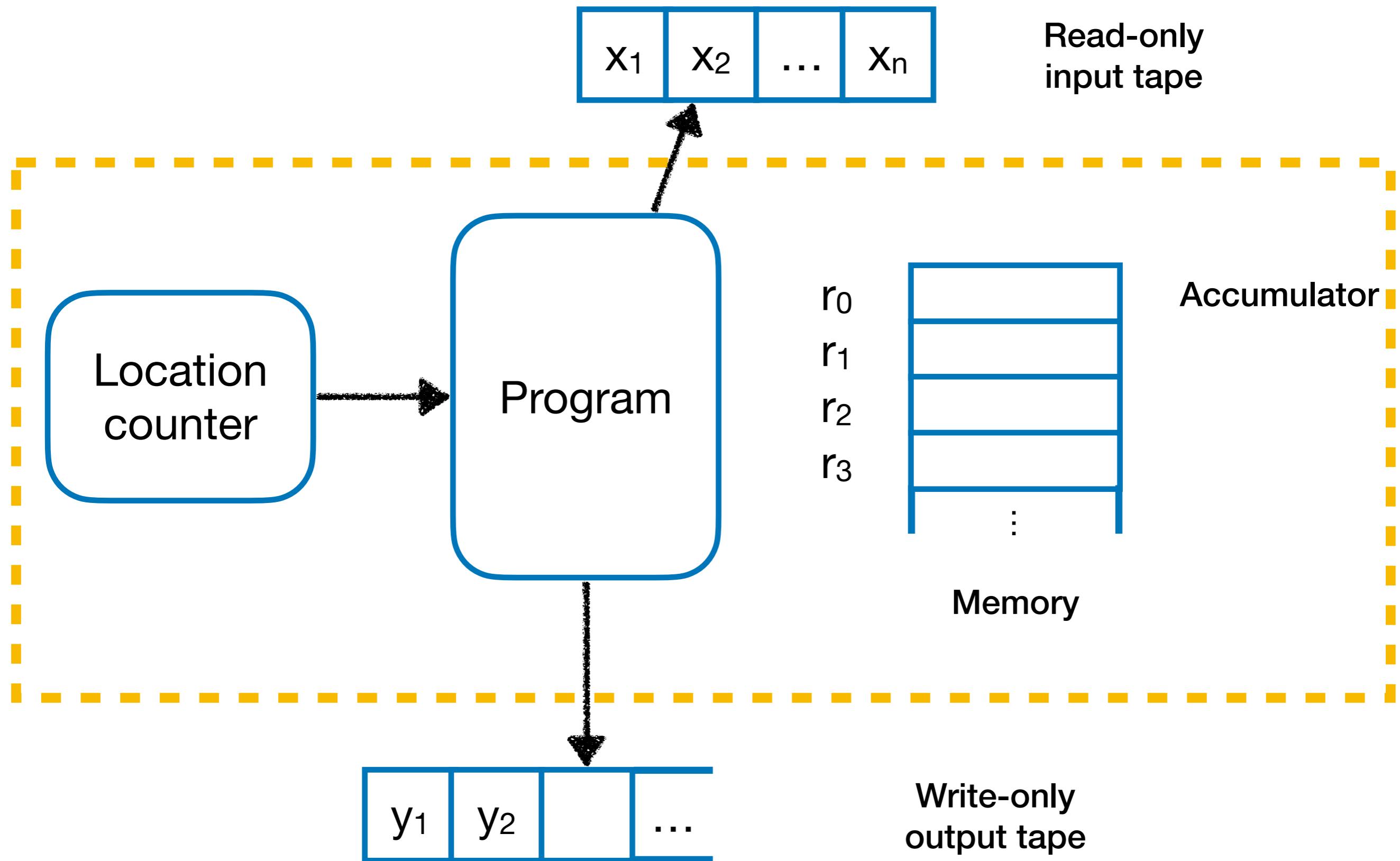
Algorithm

- Essential issues
 - Model of computation
 - Algorithm design
 - Algorithm analysis

Model of Computation

- Problems
 - Why the algorithms we learn can run almost everywhere?
 - Why the algorithms we learn can be implemented in any language?
- Machine- and language- independent algorithms, running on an abstract machine
 - Turing machine: over-qualified
 - RAM model: simple but powerful

RAM Model



The RAM Model of Computation

- Each simple operation takes one time step
 - E.g., key comparison, +/-, memory access, ...
- Non-simple operations should be decomposed
 - Loop
 - Memory
 - Memory access is simple operation
 - Unlimited memory
 - Subroutine

Tradeoff: accuracy v.s. ease of use

To Create an Algorithm

- **Algorithm design**

- Composition of simple operations, to solve an algorithmic problem

- **Algorithm analysis**

- Amount of work done / memory used
 - In the worst/average case
 - Advanced issues
 - Optimality, approximation ratio, ...

Algorithm by Example

- Algorithmic Problem 1

- Find the greatest common divisor of two non-negative integers m and n

- Algorithmic Problem 2

- Is a specific key K stored in array $E[1..n]$?

Probably the Oldest Algorithm

● Euclid Algorithm

- Find the greatest common divisor of two non-negative integers m and n

Specification

Input: non-negative integer m, n
Output: $\text{gcd}(m, n)$

Euclid algorithm

[E1] n divides m , the remainder $\rightarrow r$
[E2] if $r=0$ then return n
[E3] $n \rightarrow m; r \rightarrow n; \text{goto E1}$

Euclid algorithm - recursive version

$\text{Euclid}(m, n)$

[E1] if $r=0$ then return m
[E2] else return $\text{Euclid}(n, m \bmod n)$

Sequential Search

● Problem

- Search an array for a specific key

Specification

Input: K, E[1..n]

Output: Location of K (1,2,...,n; -1: K is not in E[])

Sequential search

```
int seqSearch(int[] E, int n, int K)
    int ans, index;
    ans = -1;
    for(index = 1; index <= n; index++)
        if(K==E[index])
            ans = index;
            break;
    return ans;
```

Algorithm Design

- Criteria

- Defining correctness

- Main challenge

- For proving correctness

- Our strategy

- Mathematical induction
 - ...

Specification

Input: non-negative integer m, n
Output: $\text{gcd}(m, n)$

Main challenge

The output is always correct, for any legal input.

Infinite possible inputs

Mathematical induction

Weak principle
Strong principle

For Your Reference

● Mathematical induction

The Weak Principle of Mathematical Induction

If the statement $p(b)$ is true and the statement $p(n-1) \Rightarrow p(n)$ is true for all $n > b$, then $p(n)$ is true for all integers $n \geq b$.

The Strong Principle of Mathematical Induction

If the statement $p(b)$ is true, and the statement $\{p(b) \text{ and } p(b+1) \text{ and } \dots \text{ and } p(n-1) \Rightarrow p(n)\}$ is true, for all $n > b$, then $p(n)$ is true for all integers $n \geq b$.

Correctness of the Euclid Algorithm

- Induction on n

- Base case

- $n = 0$: for any m , $\text{Euclid}(m, 0) = m$;
 - $n = 1$: for any m , $\text{Euclid}(m, 1) = 1$;
 - $n = 2$: ...

- Assumption

- For any $n \leq N_0$, $\text{Euclid}(m, n)$ is correct;

- Induction

- $\text{Euclid}(m, N_0+1) = \text{Euclid}(N_0+1, m \bmod (N_0+1))$;

Notes on Induction

“Notes on Structured Programming”, E.W. Dijkstra

I have mentioned mathematical induction explicitly, because it is the only pattern of reasoning that I am aware of, that eventually enables us to cope with loops and recursive procedures.

Algorithm Analysis

- Criteria
 - Performance metrics
- Worst case
 - Best case?
- Average case
 - Average cost?
- Advanced topics
 - Lower bound, optimality, ...

Algorithm Analysis

- How to measure

- Not too general
 - Giving essential indication in comparison of algorithms
- Not too precise
 - Machine independent
 - Language independent
 - Programming paradigm independent
 - Implementation independent

Algorithm Analysis

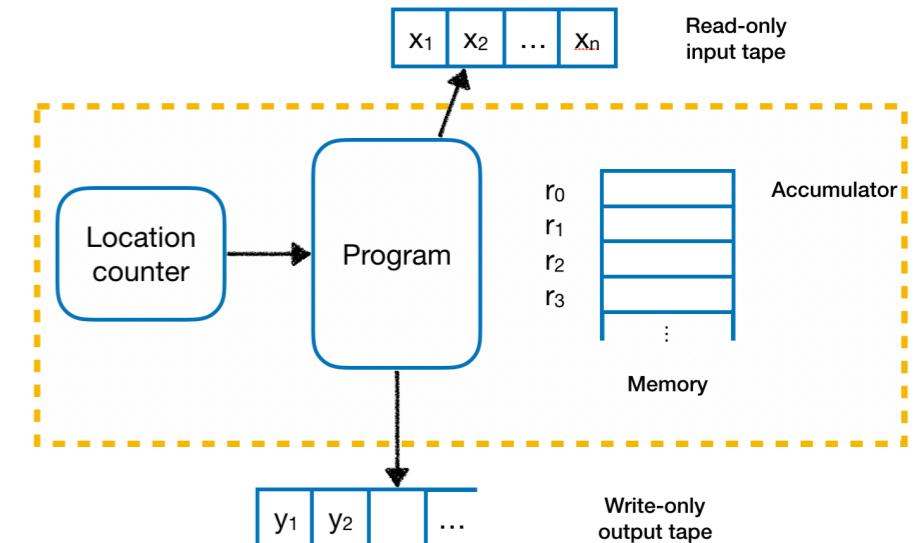
- Criteria

- Critical operation
- How many critical operation are conducted

- For example

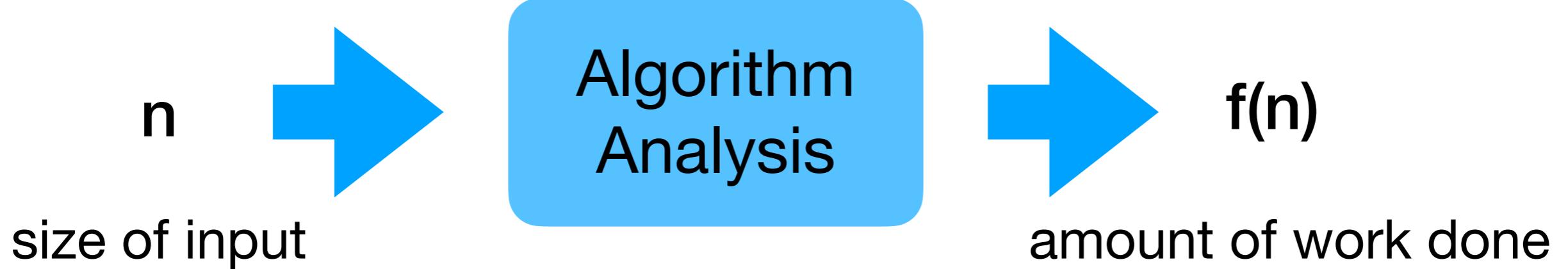
Algorithmic problem	Critical operation
Sorting, selection, searching String matching	Comparison (of keys)
Graph traversal	Processing a node/edge
Matrix multiplication	Multiplication

RAM Model



Algorithm Analysis

- Amount of work done
 - usually depends on size of the input
 - usually does not depend on size of the input only



Worst-case Complexity

- $W(n)$
 - Upper bound of cost
 - For any possible input
 - $W(n) = \max_{I \in D_n} f(I)$

Average-case Complexity

- $A(n)$

- Weighted average

$$A(n) = \sum_{I \in D(n)} Pr(I)f(I)$$

- A special case

- Average cost

- Total cost of all inputs, averaged over the input size

$$\bullet \quad A(n) = \frac{1}{|D(n)|} \sum_{I \in D(n)} f(I)$$

Average-case Cost of SeqSearch

- Case 1: K is in E[]

- Assumptions:
 - Assuming that K is in E[]
 - Assuming no same entries in E[]
 - Each possible input appears with equality (thus, K in the i^{th} location with probability $1/n$)

$$A_{\text{succ}}(n) = \sum_{i=0}^{n-1} Pr(I_i | \text{succ}) t(I_i) = \sum_{I=0}^{n-1} \frac{1}{n} (i + 1) = \frac{n + 1}{2}$$

Average-case Cost of SeqSearch

- Case 2: K may (or may not) be in E[]
 - Assume that K is in E[] with probability q

$$\begin{aligned}A(n) &= \Pr(\text{succ})A_{\text{succ}}(n) + \Pr(\text{fail})A_{\text{fail}}(n) \\&= q\frac{n+1}{2} + (1-q)n\end{aligned}$$

How to make reasonable assumptions?

Algorithm Analysis

- Advanced topics
 - Lower bound (Selection)
 - Optimality (Greedy, DP)
 - Computation complexity
 - Approximate / online / randomized algorithms

Thank you!
Q & A