Algorithm Design and Analysis

Introduction & multiplication

How was your break!?

The Big Questions

- Who are we?
- Why are we here?
- What is going on?

Who are we?

We are …

Lecture 1~12

- Instructors
  - 张宇昊 (Yuhao Zhang)
    - zhang_yuhao@sjtu.edu.cn
  - 陶表帅 (Biaoshuai Tao)
    - bstao@sjtu.edu.cn

Lecture 13~24

- We are from John Hopcroft Center for Computer Science!
  - https://jhc.sjtu.edu.cn/
  - Our Homepage:
    - www.zyhwtc.com
    - https://jhc.sjtu.edu.cn/~bstao
- Ta
  - 汪金奕 jinyi.wang@sjtu.edu.cn
  - 杨宗翰 fstqwq@sjtu.edu.cn


张宇昊

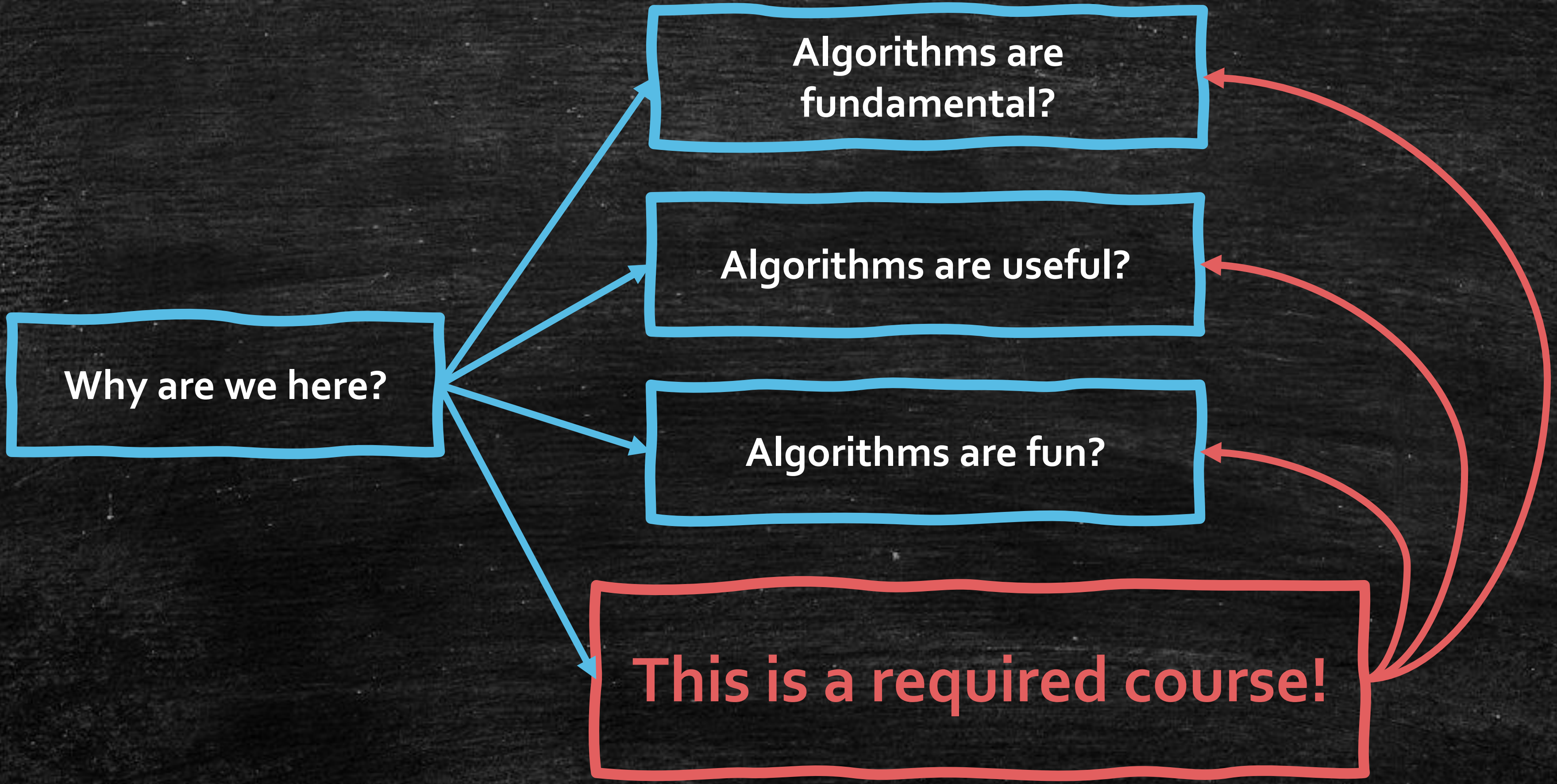

陶表帅


汪金奕

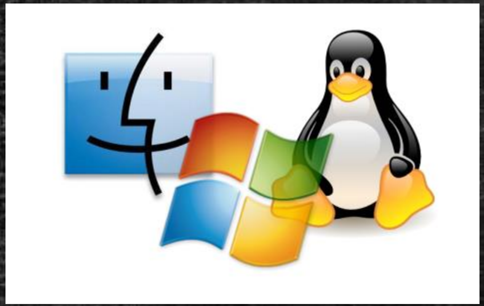

杨宗翰

Why are we here?

Algorithm!


Why are we here?
Algorithms are fundamental?
Algorithms are useful?
Algorithms are fun?
This is a required course!

Algorithms are fundamental!

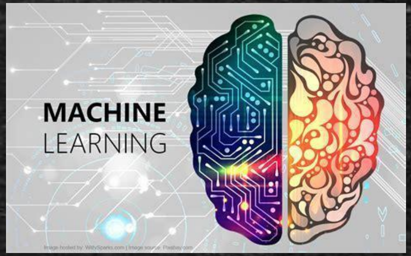


How to choose data in the cache?

Operating Systems

How to design an encoding algorithm?

Cryptography



Algorithms are useful!

- We may need to sort something?
- We may need to find the best bundle?
- …
- When the input is larger and larger,
- Algorithms are more and more important!

Algorithms are fun!

- Algorithm design is also an **art**!
- You will feel **excited** when you see a surprising algorithm!
- Your will feel **thrilled** when you have created a surprising algorithm!
- Also, many interesting research problems…

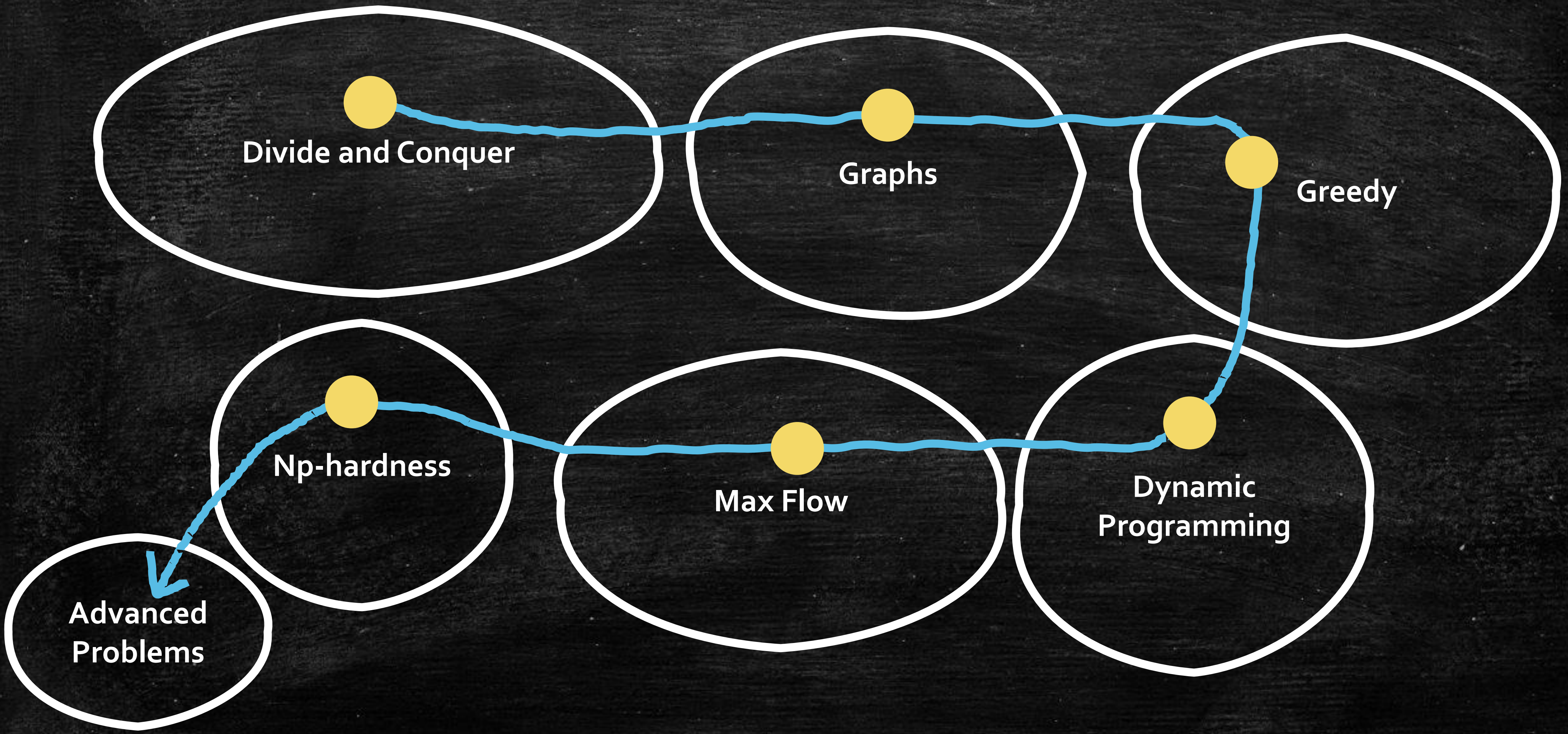
What is going on?

Course goals

- The **design** and **analysis** of algorithms
- After this course, you will
  - Think **analytically** about algorithms
  - Clearly **communicate** your algorithmic idea
  - Equip with an **algorithmic toolkit**

  - Use them **correctly**

Roadmap



Divide and Conquer
Graphs
Greedy
Dynamic Programming
Max Flow
Np-hardness
Advanced Problems

Guide questions

- Does the algorithm **work**?
- Is it **fast**?
- Can I do **better**?

How to think?

- What is work?
- What is better?
- Do we need to consider worst case?
- Is there any corner case?

Listen to my idea, it is quite intuitive! It should work if everything goes well, trust me!

Both side are necessary!

- Detail-oriented
- Precise
- Rigorous

- Big-picture
- Intuitive
- Hand-wavey

How to think in most of this course?

- We usually talk about **Exact Algorithms**.
- Dose the algorithm **work**?
  - Return the optimal/correct answer
- Is it **fast**?
  - Time complexity
  - Worst case
- Can I do **better**?
  - More efficient
  - Better time complexity

Aside the course……

- What if the problem is so hard to get the solution?
  - Np-hard problems: take too long time
  - Online problems: not enough information

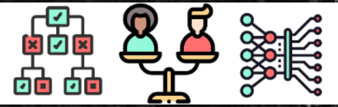
$$(x \text{ OR } y \text{ OR } z) \text{ AND } (x \text{ OR } \bar{y} \text{ OR } z) \text{ AND } (x \text{ OR } y \text{ OR } \bar{z}) \text{ AND } (x \text{ OR } \bar{y} \text{ OR } \bar{z}) \text{ AND } (\bar{x} \text{ OR } y \text{ OR } z) \text{ AND } (\bar{x} \text{ OR } \bar{y} \text{ OR } \bar{z})$$

SAT Problem

Online Matching

- What if a more efficient algorithm is not better?
  - More efficient → make private data public
  - More efficient → focus on the majority population?

Data Privacy



Fairness

- What if you can not control player's behavior?
  - Auction
  - Public resource allocation

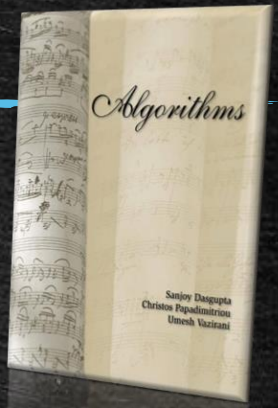
Auction

Public Resources

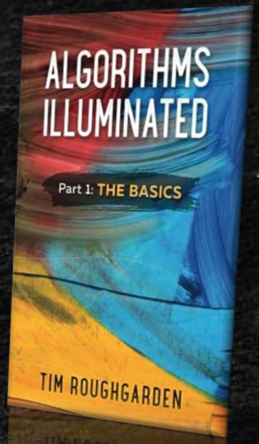
About the course?

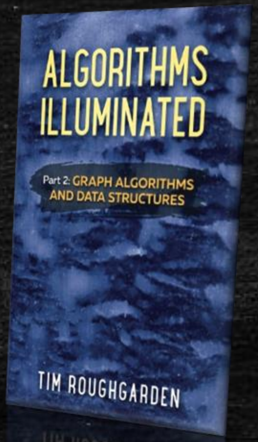
References (optimal)

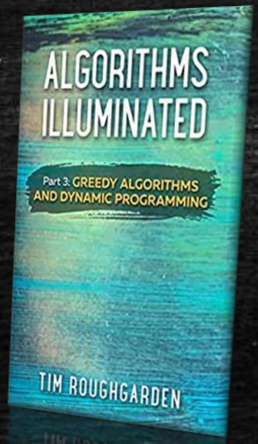
- **Algorithms by Dasgupta, Papadimitriou, Vazirani**



- **Algorithms Illuminated, Vols 1,2 and 3 by Tim Roughgarden**







Homework

- Homework: $\mathbf{70\%}$
  - 12 (6 writing + 6 programming) homework: $\boldsymbol{a \leq 60\%}$
  - 1 midterm (in-class): $\boldsymbol{b \leq 20\%}$
- 1 final exam: $\boldsymbol{c \leq 30\%}$
- Overall: $\boldsymbol{\min\{a + b, 70\} + c}$
- We encourage **discussion**, but please try them on your **own** before discussion, and conclude them on your **own** after discussion.

Talk to us and each other!

- You can discuss with us at office hours.
  - Question: I do not know how to do it?
  - Question: This is my approach, but I got a stuck here…
- Office hours
  - Yuhao (Fri 4:00~5:00pm)
  - Biaoshuai (Mon 3:00~4:00pm)
  - Jinyi (Fri 9:00~10:00am)
  - Zonghan (Thu 4:00~5:00pm)
- Wechat group
  - Check **CANVAS**

Policy

- We encourage discussion on homework, but you should **write down** your solution **on your own**.
- You must **cite** all collaborators, as well as all sources used (e.g., online materials).
- Late policy
  - Within 3 days: **50%** of your score
  - Out of 3 days: **0%**
  - Special Issue

Feedback

- It's my **first course**, so please tell me
  - The **pace** of the lecture
  - The **difficulty** of the homework
  - The **tpyos** in the sldies

Today

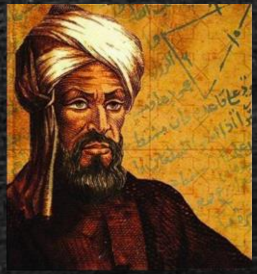
Integer Multiplication

Today's goal

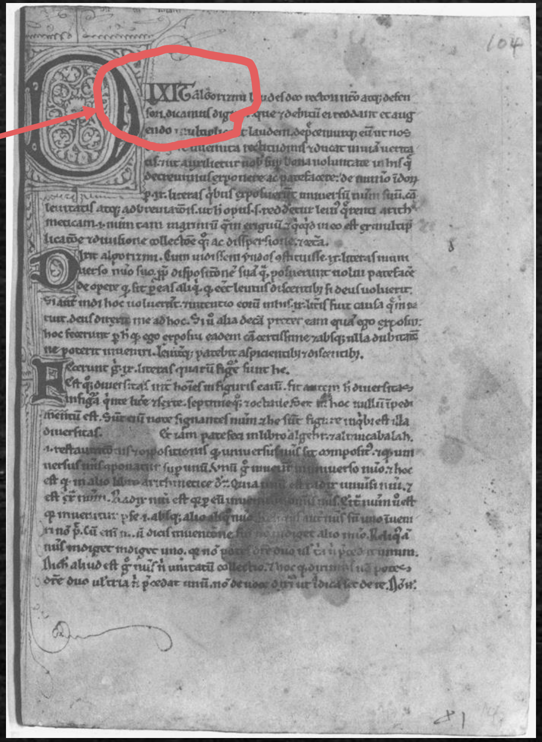
- Karatsuba Integer Multiplication
- Algorithmic Technique
  - Divide and conquer
- Algorithmic Analysis tool
  - Intro to asymptotic analysis

Start at very beginning

- al-Khwarizmi



- Dixit algorizmi



- “Algorisme” [old French]
  - Arabic number system
  - “Algorithm”

Integer Multiplication

- How to calculate 44×34

$$\begin{array}{r} 44 \\ \times\ \ 34 \\ \hline \end{array}$$

- How to calculate $123555589 \times 987555321$

$$\begin{array}{r} 123555589 \\ \times\ \ 987555321 \\ \hline \end{array}$$

How fast is it?

$$
\begin{array}{r}
\overbrace{123555589124435234523465324}^{n} \\
\times\ 875553211231231231231233123 \\
\hline
\end{array}
$$

How many 1-digit operation we need to make?

Roughly
- n^2 multiplication
- n^2 addition for carries
- $2n$ addition finally

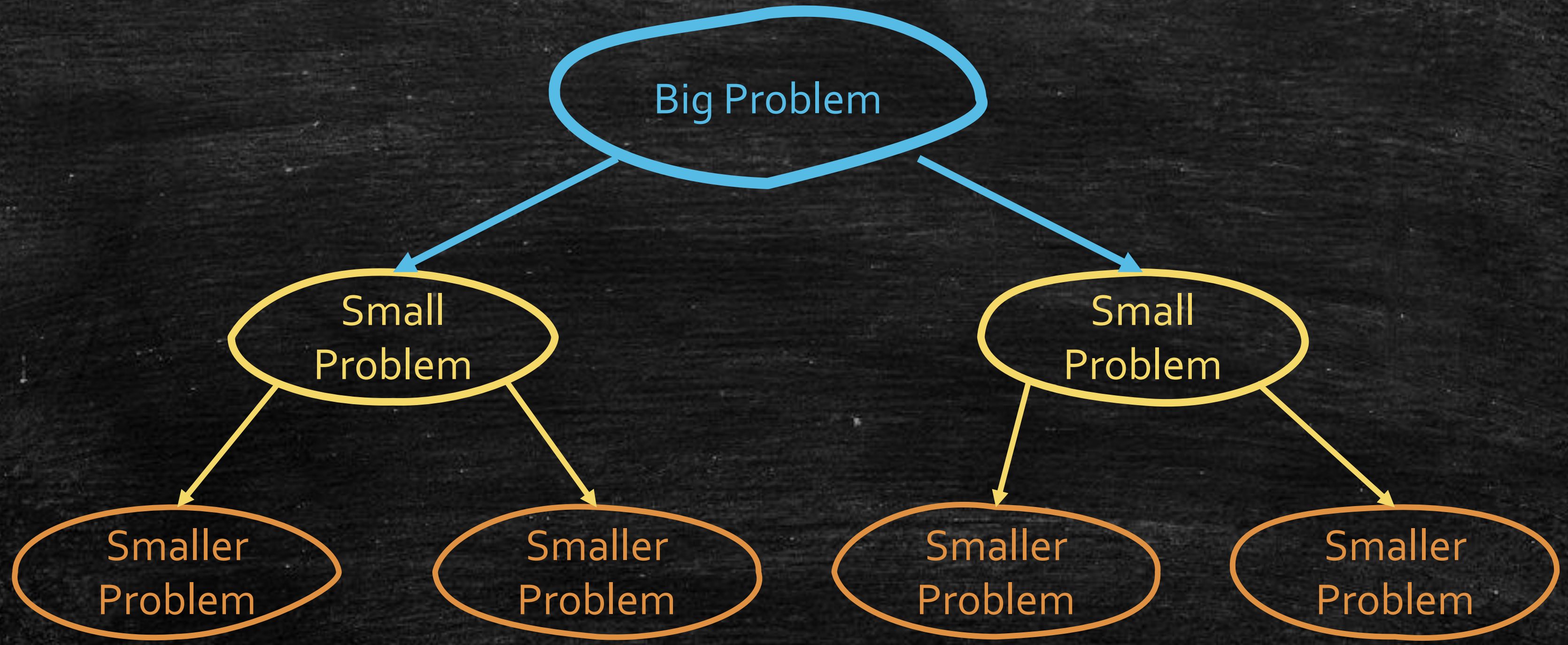
$O(n^2)$

Can we do better?

Let us buy our first tool!

Divide and conquer

Divide and Conquer



Divide and conquer for multiplication

- 1234×5678
- $1234 = 12 \times 100 + 34$
- $1234 \times 5678 = (12 \times 100 + 34)(56 \times 100 + 78)$
  $= (12 \times 56) \cdot 10000 + (12 \times 78 + 34 \times 56) \cdot 100$
  $+ 34 \times 78$
- 1 four-digit → 4 two-digit

Generally?

- Can we make it generally?
- Two n digit multiplications, suppose n is even
- Design a recursive algorithm for n, suppose n is 2's power.
- $xy = \left(a \cdot 10^{\frac{n}{2}} + b\right)\left(c \cdot 10^{\frac{n}{2}} + d\right)$

  $= ac \cdot 10^{n} + (ad + bc) \cdot 10^{\frac{n}{2}} + bd$

Running time, analytically

- Main question: **Is it better than before?**
  - ~~**Yes! Because we learn it in SJTU!**~~
  - how many 1-digit multiplications we need for 1n-digit multiplication?
    - A: n^2; B: n^3; C:n; D: $n\ \log n$;
    - Run the algorithm for 1234×5678, how many 1-digit multiplications we need?
    - how many 1-digit multiplications we need for 1 8-digit multiplication?

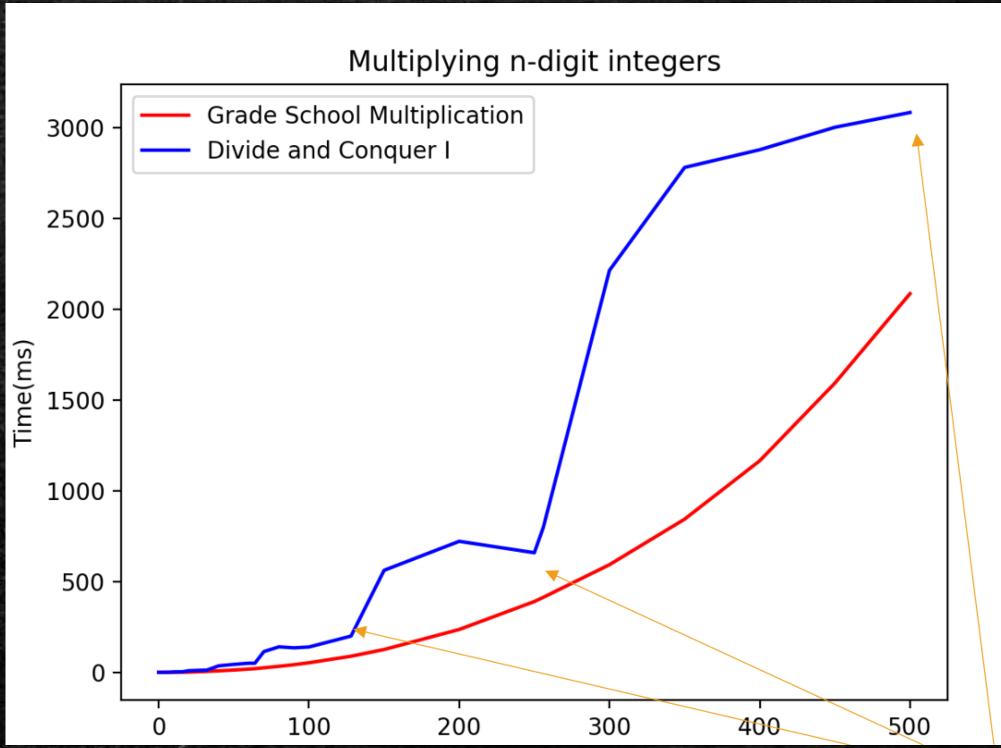
Analysis

- Claim: we need n^2 1-digit multiplications for 1 n-digit multiplication.
- How many levels we need?
  - $\log_2 n$
- How many multiplications we need in level $t = \log_2 n$?
  - Level 0: 1 $n \times n$
  - Level 1: 4 $\frac{n}{2} \times \frac{n}{2}$
  - Level 2: 16 $\frac{n}{4} \times \frac{n}{4}$
  - Level t: 4^t 1×1
- Conclusion: $4^{\log_2 n} = n^2$

It is just an analysis!

Experiments

- Claim: the grade school multiplication is better!



What's wrong?

- $xy = \left(a \cdot 10^{\frac{n}{2}} + b\right) \times \left(c \cdot 10^{\frac{n}{2}} + d\right)$
  $= ac \cdot 10^n + (ad + bc) \cdot 10^{\frac{n}{2}} + bd$
- What do we need?
  - ac
  - $\boldsymbol{ad + bc}$
  - bd
- What do we calculate
  - ac
  - $\boldsymbol{ad}$
  - $\mathbf{bc}$
  - bd

Karatsuba Algorithm

Improve!

- What do we need?
  - ac
  - $\boldsymbol{ad + bc}$
  - bd
- How to get $\boldsymbol{ad + bc}$ without $\boldsymbol{ad}$ and $\boldsymbol{bc}$?
- Solution:
  - Calculate: ac, bd
  - One more multiplication: $\mathrm{z} = (a+b)(c+d)$
  - Get $ad + bc = (a+b)(c+d) - ac - bd$
  - $x \times y = \left(a \cdot 10^{\frac{n}{2}} + b\right) \times \left(c \cdot 10^{\frac{n}{2}} + d\right)$

$$= ac \cdot 10^n + (ad + bc) \cdot 10^{\frac{n}{2}} + bd$$

$$= ac \cdot 10^n + (z - ac - bd) \cdot 10^{\frac{n}{2}} + bd$$

Improve!

- What is the difference?
  - We now calculate
    - ac
    - $\boldsymbol{z = (a + b)(c + d)}$
    - bd
  - One n-digit → Three $\frac{n}{2}$-digit

Make a guess!

How fast is it?

Is it fast?

- Claim: we need $n^{1.6}$ 1-digit multiplication for 1 n-digit multiplication.
- How many levels we need?
  - $\log_2 n$
- How many multiplications we need in level t?
  - Level 0: 1 $n \times n$
  - Level 1: 3 $\frac{n}{2} \times \frac{n}{2}$
  - Level 2: 9 $\frac{n}{4} \times \frac{n}{4}$
  - Level t: 3^t 1×1
- Conclusion: $3^{\log_2 n} = n^{\log_2 3} \approx n^{1.6}$

What if n is **not** $2's$ power?

Can we do better **again**?

Better algorithms

- Toom-Cook (1963): Breaking into size $\frac{n}{3}$-size problems make it better! → $O(n^{1.465})$
- Think:
  - how to break $n \times n$ into $5\ \frac{n}{3} \times \frac{n}{3}$?
  - Given it is true, why it is $n^{1.465}$?

$$\log^* n := \begin{cases} 0 & \text{if } n \le 1; \\ 1 + \log^*(\log n) & \text{if } n > 1 \end{cases}$$

- Schonhage-Strassen (1971): $O(n \log n \log\log n)$
- Furer (2007): $O(n \log n \log^* n)$
- Harvey and van der Hoeven (2019): $O(n \log n)$

Our work is expected to be the end of the road for this problem, although we don't know yet how to prove this rigorously.

What about matrix?

- How to multiply two matrices
- $\begin{bmatrix} 2 & 9 \\ 7 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 \times 1 + 9 \times 3 & 2 \times 2 + 9 \times 4 \\ 7 \times 1 + 5 \times 3 & 7 \times 2 + 5 \times 4 \end{bmatrix} = \begin{bmatrix} 29 & 40 \\ 22 & 34 \end{bmatrix}$
- $Z = XY$
- $z_{ik} = \sum_{1 \le j \le n} x_{ij} y_{jk}$
- How many integer multiplications?
  - n^2 entries of Z to calculate
  - Each takes n multiplications
  - Totally n^3
- What about running time?

Word Ram model?
Turing model?

How to divide and conquer?

Divide and conquer

- Key fact: If $X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$.
  - $\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & \mathrm{F} \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & BF + DH \end{bmatrix}$
- How to divide and conquer?
  - 1 n-size multiplication → 8 $\frac{n}{2}$-size multiplications
    - $AE, BG, AF, BH, CE, DG, BF, DH$
  - How many integer multiplications?
  - $8^{\log_2 n} = n^3$
  - **The same problem as before!**

Do you have any approach?

Strassen's magical idea

- $P_1 = A(F - H)$
- $P_2 = (A + B)H$
- $P_3 = (C + D)E$
- $P_4 = D(G - E)$
- $P_5 = (A + D)(E + H)$
- $P_6 = (B - D)(G + H)$
- $P_7 = (A - C)(E + F)$

- $XY = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & \mathrm{F} \\ G & H \end{bmatrix}$

$$= \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & BF + DH \end{bmatrix}$$

$$= \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

- How many integer multiplications now?
- $7^{\log_2 n} = n^{\log_2 7} \approx n^{2.81}$

Goals!

- Course goals
  - Think **analytically** about algorithms
  - Clearly **communicate** your algorithmic idea
  - Equip with an **algorithmic toolkit**
- Today's goals
  - Karatsuba Integer Multiplication
  - Algorithmic Technique
    - Divide and conquer
  - Algorithmic Analysis tool
    - Intro to asymptotic analysis

How about the pace today?

Next time

- More divide and conquer

- Before next time
  - Think the questions in the slides.
  - Join the wechat group!
  - Try the Online Judge System!

Welcome to discuss research problems with us!