CS473 Algorithms: Lecture 1 (2022-01-18)

logistics: - pset 0 out Friday 5pm, due 1 week after

- signup - piazza
  - gradescope

today: - introduction
- divide and conquer
  - integer multiplication

=

lecture: TR 14-15:15 Siebel 1404

staff: instructor: Prof Michael A. Forbes (miforbes) T 3:30, TBA

TA: Christian Howard (howard28) TBA

TA: Pooja Kulkarni (poojark2) TBA

resources: webpage: - courses.engr.illinois.edu/cs473/sp2022

- full policy details

calendar: - lectures - topics
  - materials
  - readings

- psets

forum (piazza):

- course announce
- peer discussion
- contact staff

submission (gradescope):

- course work submission/return/regrades
- gradebook

course materials:

- lectures: in-person
  except for first week
  on zoom
- lecture material
  - slides/handwritten
  - videos
- textbooks - suggested reading for each lecture
  ↳ 90% from Kleinberg Tardos

thm: reading course webpage makes you more knowledgeable

pf: by authority

cor: ______ after ______ even more ______

grades:
- psets (25%)
  - 12 psets, 3 problems each
  - out/due F17
  - no late psets
    - two lowest psets scores dropped
  - pset groups
    - pset0 done individually
    - psetN, $N \geq 1$ can be done in groups ≤ 3
    - integrity
- exams (45%) - $2 \times 22.5\%$
  - noncumulative
  - dates 2022-02-28 19-21:30
    2022-04-11 19-21:30
- final (30%) - cumulative

prereq:
formal:
- CS173 (discrete maths)
- CS225 (data structures)
- CS374 (algo, models of computation)

informal:
- formal proofs
- basic algo
  - data structures
  - graph algo
- probability
- models of computation

Q: why this course?

motivation: google is really useful

- maps
- flights
- search

Q: how does google do it?

A: algorithms!

Q: can algorithms do everything?

A: no

fact (cs 374): exist ~~computational~~ problems that [cannot] be solved by computers

↳ undecidable problems

fact (cs 579): exist [solvable] ________ [efficiently] ______

Q: which problems can be solved efficiently?

A: no idea

this course: fundamental algorithmic paradigms for designing efficient algo

- divide and conquer
- dynamic programming
- cut and flows
- linear programming
- NP-completeness
- approximation algorithms

theme: road to efficient algo is winding, long, and filled w/ math divide and conquer

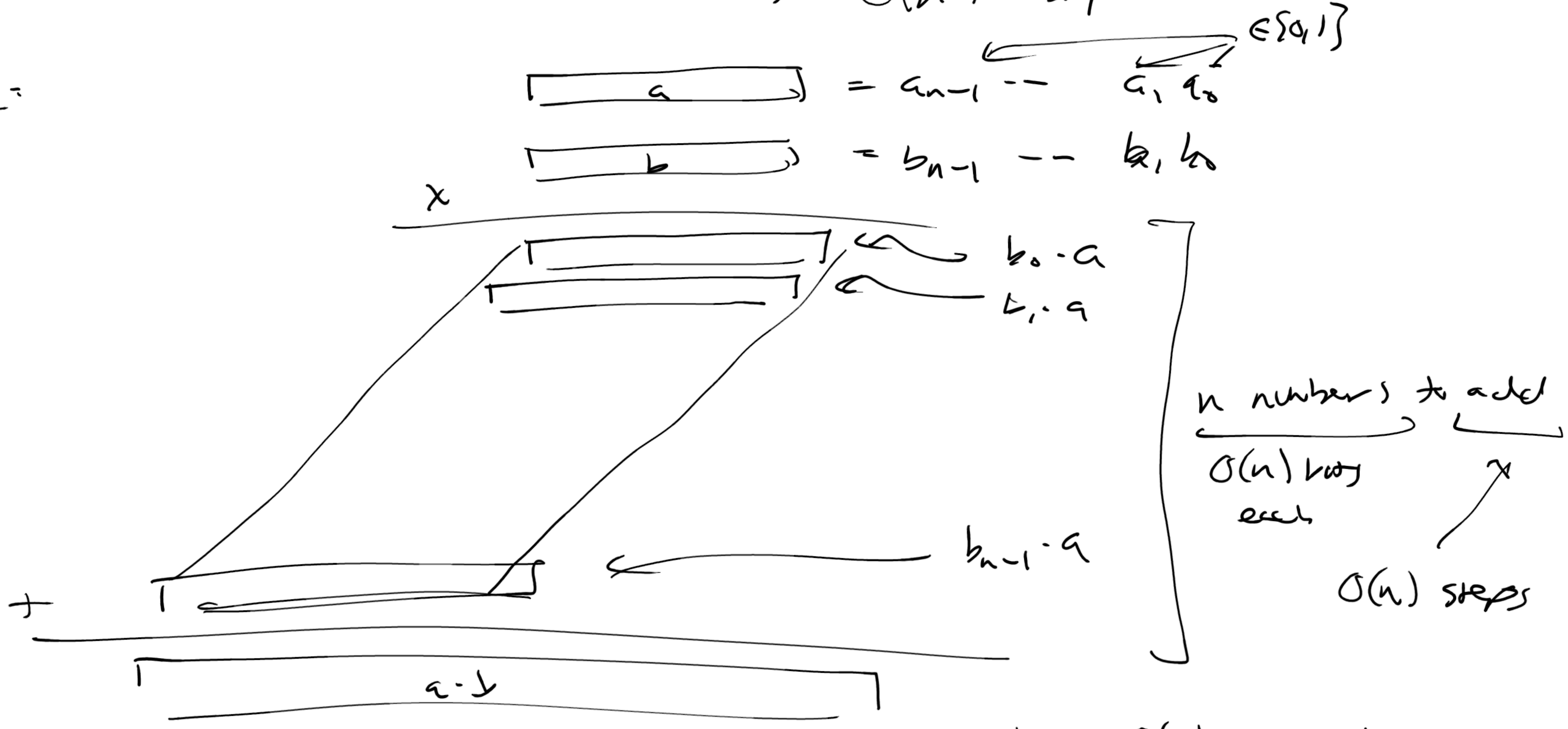
idea:

recall grade school multiplication of two n-bit numbers takes $O(n^2)$ steps

eg:

```
      111    = 7
  x   101    = 5
  -------
      111
     000
 +  111
  -------
   100011    = 35
   32 16 8 4 2 1
```

eg:



$\Rightarrow n \cdot O(n) = O(n^2)$ steps

Q: Can we do better?

lem: multiplication of two n-bit numbers in $O(n^2)$ steps, $\boxed{\text{with}}$ divide and conquer

pf: a, b n-bit numbers $a = a_1 \cdot 2^{n/2} + a_0$ w/ a_1, a_0 $n/2$ bits

$[\ a\ ] = [\ a_1 \mid a_0\ ]$

$b = b_1 \cdot 2^{n/2} + b_0$ w/ b_1, b_0 ——

$a \cdot b = (a_1 \cdot 2^{n/2} + a_0)(b_1 2^{n/2} + b_0)$ $\boxed{\text{shifts}}$ multiplication, can be done in $O(n)$ time

$= \underbrace{a_1 b_1}\, 2^n + (\underbrace{a_0 b_1} + \underbrace{a_1 b_0})\, 2^{n/2} + \underbrace{a_0 b_0}$] conquer - 3 additions - 2 shifts

$O(n)$ steps

4 multiplications in $n/2$-bit numbers] divide

correctness: clear

complexity: $T(n)$ = worst case time of any n-bit mult

$\leq 4 \cdot T(n/2) + O(n)$

$\leq O(n^2)$

Q: can we do better?

thm [Karatsuba 60]. ________ in $\underbrace{O(n^{\log_2 3})}_{= O(n^{1.584\ldots})}$ steps.

pf: $a = a_1 \cdot 2^{n/2} + a_0$

$b = b_1 \cdot 2^{n/2} + b_0$

$a \cdot b = a_1 b_1 2^n + (a_0 b_1 + a_1 b_0) 2^{n/2} + a_0 b_0$

above: use 4 recursive calls to compute 3 numbers $a_1 b_1$, $a_0 b_1 + a_1 b_0$, $a_0 b_0$

idea: use 3

lem: $(a_1 - a_0)(b_1 - b_0) = a_1 b_1 + a_0 b_0 - (a_0 b_1 + a_1 b_0)$

lem: $\in (-2^{n/2}, 2^{n/2})$, so this is a $n/2$-bit multiplication (after adjusting sign)

also: (1) recursively compute $a_1 b_1$, $a_0 b_0$, $(a_1 - a_0)(b_1 - b_0)$

(2) compute $a_0 b_1 + a_1 b_0$ via

(3) compute $a \cdot b$ via

correctness: clear

complexity: $T(n) \in 3 \cdot T(n/2) + O(n) \subseteq O(n^{\log_2 3})$ ∎

rmks:

- $\Omega(n^2)$ conjectured necessary by Kolmogorov 60
- Karatsuba 60 disproved this
- Toom 63 / Cook 66:

  split n-bit number into $k \geq 2$ parts

  $\Rightarrow$ multiplication in $n^{1+O(1/\lg k)}$

  for $k \leq O(1)$
- Gauss 1800's / Cooley Tukey 65 / Schonhage Strassen 71:

  multiplication via Fast Fourier transform

  in $O(n \lg n \lg\lg n)$ steps
- Fürer 07 $O(n \lg n \, 2^{O(\lg^* n)})$

  iterated logarithm
- Harvey - van der Hoeven 19 $O(n \lg n)$

Q do better?

A. not believed likely

today: - introduction

- divide and conquer

  - integer multiplication

    Karatsuba's alg.

next lecture: divide and conquer

logistics: - pset 0 out F 1/7

  due 1 week later

- syncup - piazza
  - gradescope