Advanced Algorithms

Lecture 3: Divide and Conquer

Announcements

- HW 1 is out! (due next Friday — Sep 6th)
- Piazza
- **<u>Late policy for HWs</u>**
- TA office hours
- *Attendance in class, videos!*
- <u>Midterm exam:</u> Thu, October 3 (in class)

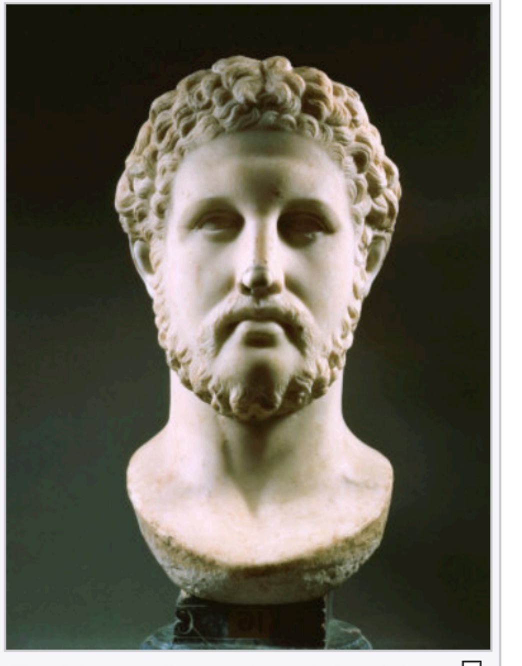
Last week

- Basic example of analysis by induction — bubble sort, binary search
- Data structures — a one-class intro
- “API” — what is being stored? what are the operations? costs?
- Amortized analysis (total time for N operations is $O(N)$)
- Notes on course page — pls read/comment!

Today

Basic paradigm #1: Divide and conquer

Divide and rule (Latin: *divide et impera*), or **divide and conquer**, in politics and sociology is gaining and maintaining power by breaking up larger concentrations of power into pieces that individually have less power than the one implementing the strategy.



Tradition attributes the origin of the motto to Philip II of Macedon: Greek: διαίρει καὶ βασίλευε *diaírei kài basíleue*, in ancient Greek: «divide and rule»

The use of this technique is meant to empower the sovereign to control subjects, populations, or factions of different interests, who collectively might be able to oppose his rule. Niccolò Machiavelli identifies a similar application to military strategy, advising in Book VI of *The Art of War* (1521)[1] (*L'arte della guerra*):[2] a Captain should endeavor with every art to divide the forces of the enemy. Machiavelli advises that this act should be achieved either by making him suspicious of his men in whom he trusted, or by giving him cause that he has to separate his forces, and, because of this, become weaker.

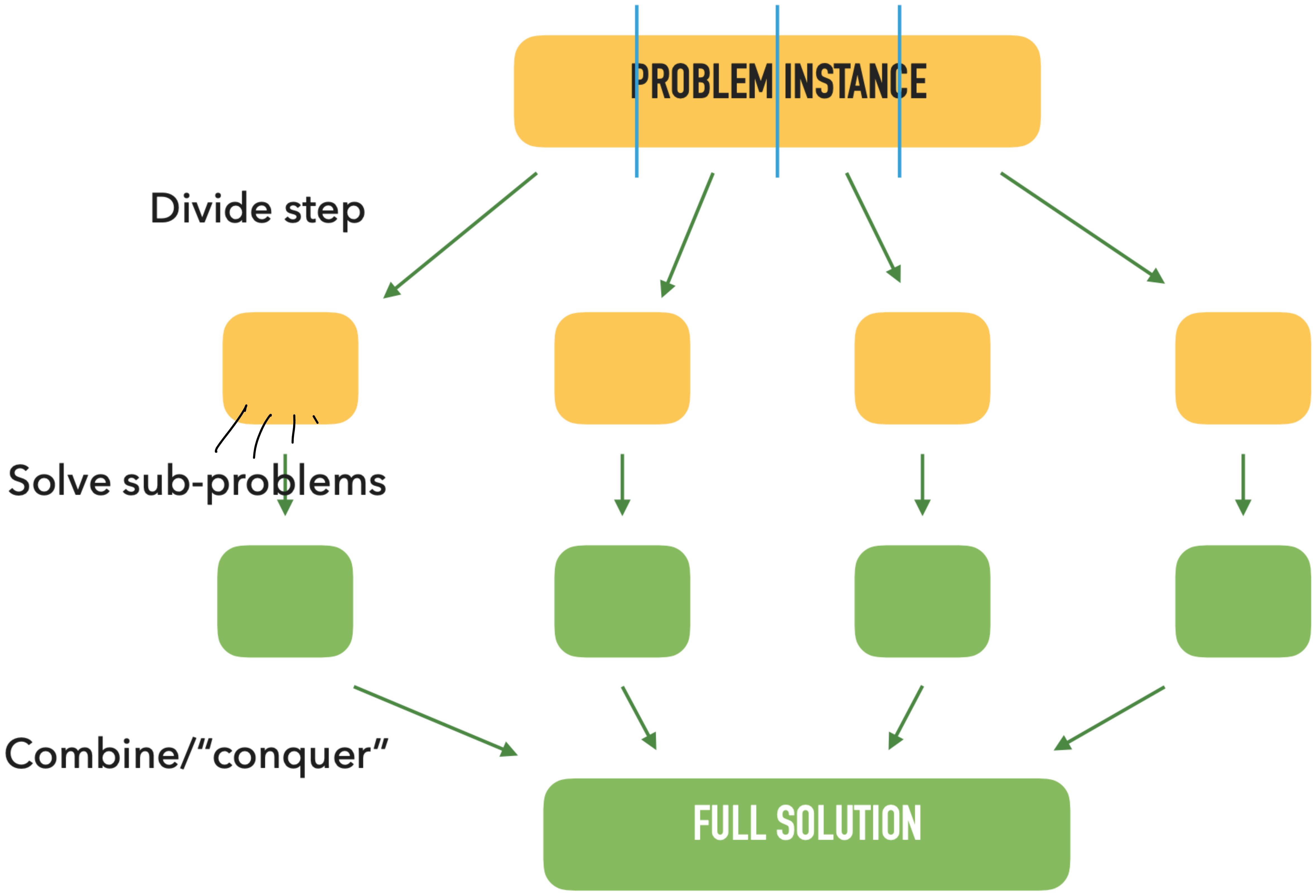
The maxim ***divide et impera*** has been attributed to Philip II of Macedon. It was utilised by the Roman ruler Julius Caesar and the French emperor Napoleon (together with the maxim *divide ut regnes)*

The strategy, but not the phrase, applies in many ancient cases: the example of Aulus Gabinius exists, parting the Jewish nation into five conventions, reported by Flavius Josephus in Book I, 169–170 of *The Jewish War* (*De bello Judaico*).[3] Strabo also reports in *Geographica*, 8.7.3[4] that the Achaean League was gradually dissolved under the Roman possession of the whole of Macedonia, owing to their not dealing with the several states in the same way, but wishing to preserve some and to destroy others.

The strategy of division and rule has been attributed to sovereigns, ranging from Louis XI of France to the House of Habsburg. Edward Coke denounces it in Chapter I of the Fourth Part of the *Institutes of the Lawes of England*, reporting that when it was demanded by the Lords and Commons what might be a principal motive for them to have good success in Parliament, it was answered: "*Eritis insuperabiles, si fueritis inseparabiles. Explosum est illud diverbium: Divide, & impera, cum radix & vertex imperii in obedientium consensu rata sunt.*" [You would be invincible if you were inseparable. This proverb, Divide and rule, has been rejected, since the root and the summit of authority are confirmed by the consent of the subjects.] On the other hand, in a minor variation, Sir Francis Bacon wrote the phrase "separa et impera" in a letter to James I of 15 February 1615. James Madison made this recommendation in a letter to Thomas Jefferson of 24 October 1787,[5] which summarized the thesis of *The Federalist#10*:[6] "Divide et impera, the reprobated axiom of tyranny, is under certain (some) qualifications, the only policy, by which a republic can be administered on just principles." In *Perpetual Peace: A Philosophical Sketch* by Immanuel Kant (1795), Appendix one, *Divide et impera* is the third of three political maxims, the others being *Fac et excusa* (Act now, and make excuses later) and *Si fecisti, nega* (If you commit a crime, deny it).[7]

Elements of this technique involve:

- creating or encouraging divisions among the subjects to prevent alliances that could challenge the sovereign
- aiding and promoting those who are willing to cooperate with the sovereign
- fostering distrust and enmity between local rulers
- encouraging meaningless expenditures that reduce the capability for political and military spending


PROBLEM INSTANCE
Divide step
Solve sub-problems
Combine/"conquer"
FULL SOLUTION

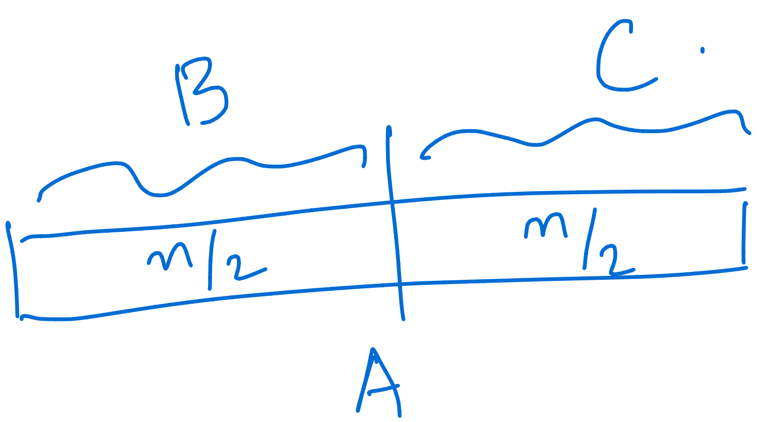
Today's plan

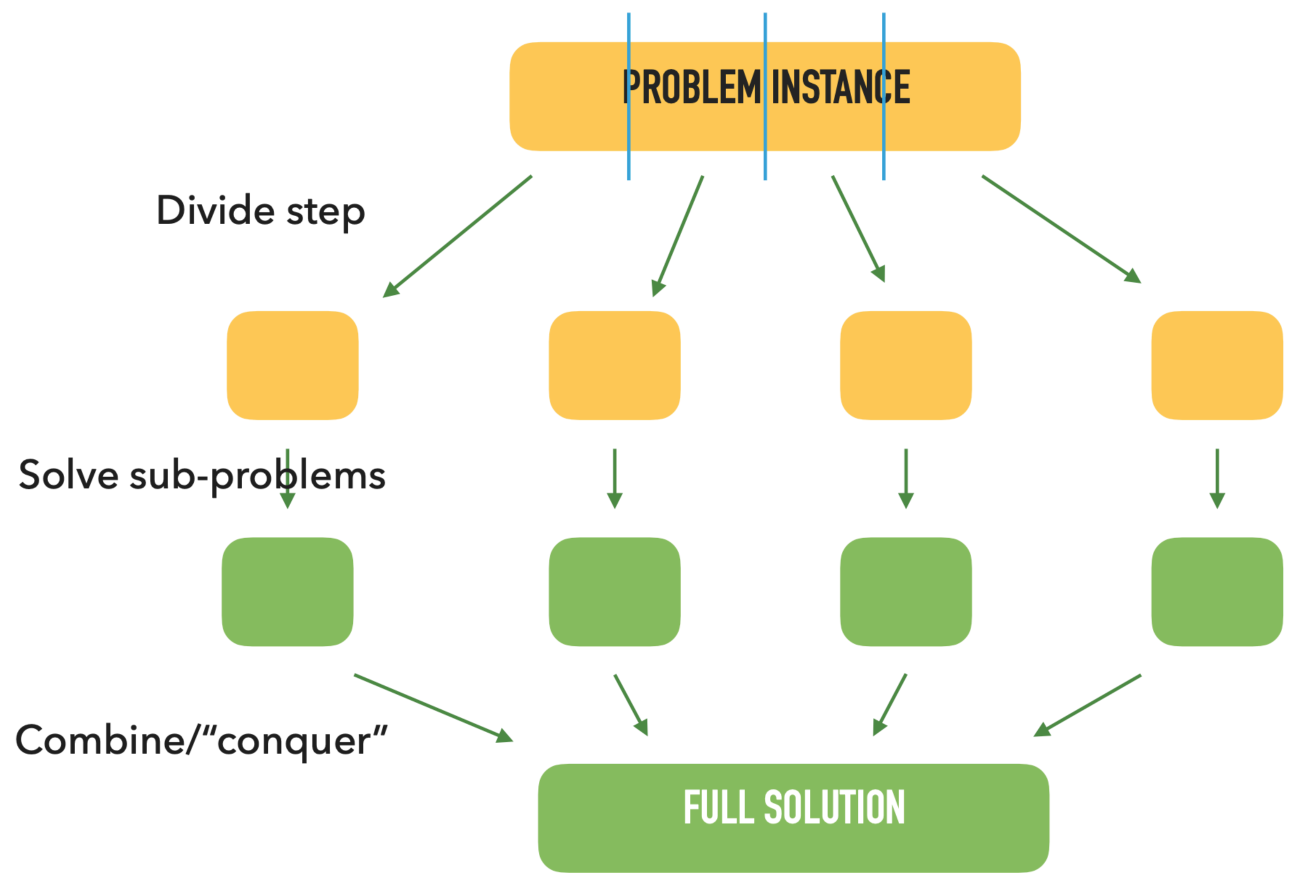
- Examples — divide and conquer may(not) be easy to "spot"
- How to analyze divide and conquer algorithms
    - correctness (typically done via induction).
    - running time/complexity (recurrences)

Example 1: sorting

Problem: given an array A of size n, place elements in increasing (non-decreasing) order.

- Divide and conquer?



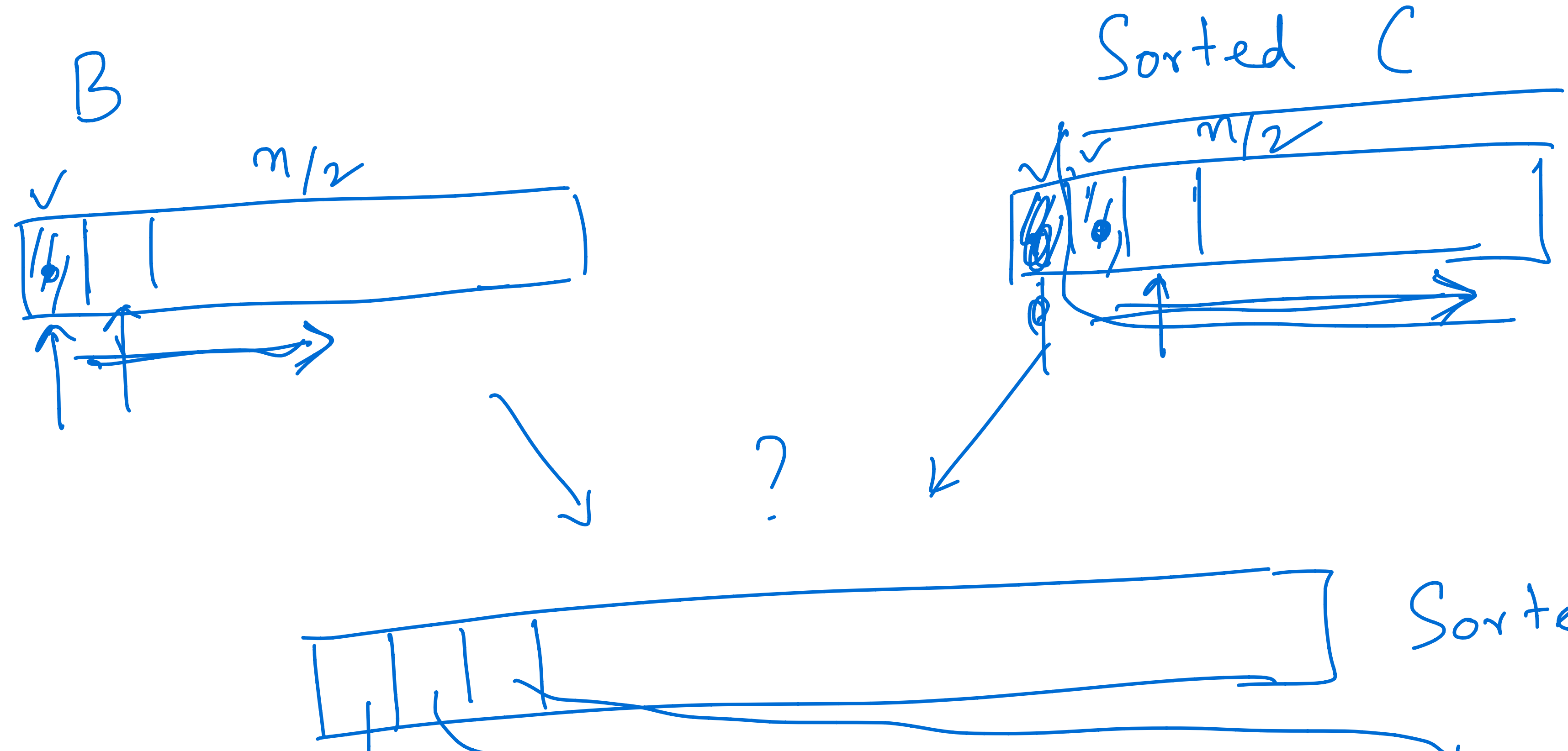


B : | 3 | 4 | 7 | 8 |

C : | 1 | 2 | 9 | 11 |

Conquer step

Running time $= O(n)$



$\min\{B[0], C[0]\}$ $\quad \min\{B[0], C[1]\}$ $\quad \min\{B[0], C[2]\}$

→ If we have two sorted lists B & C, then the minimum element of $B \cup C$ is $\min\{B[0], C[0]\}$.

– So we delete that min element (from B or C) & recurse.

<u>Merge(B, C)</u>: produces a new array A.

Until B and C are both empty:

- add min(B[0], C[0]) to A & remove it from corr. array.
- if either B is empty or C is empty, treat B[0] or C[0] as ∞.

Overall procedure

sort $\left(\text{array } A[0, \ldots, n-1]\right)$:

if $n=1$, return the array! $\longrightarrow O(1)$

create sub-arrays $B = A\left[0, \ldots, \frac{n}{2}\right]$ and
$C = A\left[\frac{n}{2}+1, \ldots, n-1\right]$ $\longrightarrow O(n)$

(recursively) $\boxed{\text{sort}(B)\ ;\ \text{sort}(C)}$;

Run "merge" procedure on sorted B, C. $\longrightarrow O(n)$

Proof by induction: — sort(A) gives right answer when $n=1$

— Suppose sort(A) returns right answer for all sizes $< n$, for any $n \geq 2$.

then sort(A) " " for n.

Correctness

- Template for recursive algorithms:
  - ✓ show base cases are correct ✓
  - assume recursive steps give right answer
  - ✓ show that "combination" is correct

Claim: if B & C are sorted & we run the merge procedure, the resulting array is the sorted version of A.

Lemma 1: all the elements of B ∪ C appear in A.

Lemma 2: Elements in A will be in increasing order.

Proof is easy because in each step, we pick the smallest elt. of the "current" B ∪ C.

Proof of Lemma 1:

- Once an element of B or C is written to A, it gets deleted (so it never shows up again).
- The procedure runs until B & C become empty $\Rightarrow$ all elements in $B \cup C$ show up at least once.

Running time

$$\text{time}(A) := 1 + n + \underbrace{\text{time}(B) + \text{time}(C)}_{\text{arrays of size } \frac{n}{2}} + n$$

A: array of size n

$T(n)$: time taken for sort( ) on an array of length $= n$

$$T(n) = 2n+1 + 2T\left(\frac{n}{2}\right)$$

$$\underline{T(n) = 2T\left(\frac{n}{2}\right) + O(n)}.$$

$\hookrightarrow$ recurrence relations.

$$T(n) = T(n-1) + T\left(\frac{n}{2}\right) + 1$$

Recurrences

$$\boxed{T(n) = 2T\left(\frac{n}{2}\right) + c \cdot n}$$

- General form: f(n) = function (n, f(1), ..., f(n-1))
- Finding a "closed form" can be challenging
  - no silver bullet
  - many "techniques" — recursion tree, plug-and-chug, guess-and-prove, *master theorem*, Akra-Bazzi theorem, ...
  - personal favorites...

$T(1) = 1.$

Tree: T(n) = 2T(n/2) + cn

$T(n)$

$\longrightarrow T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + c \cdot n \longrightarrow \boxed{cn}$

$T\left(\frac{n}{4}\right) + T\left(\frac{n}{4}\right) + \boxed{c \cdot \frac{n}{2}} \quad T\left(\frac{n}{4}\right) \quad T\left(\frac{n}{4}\right) \quad \boxed{c \cdot \frac{n}{2}} \longrightarrow$ total $= cn$

2 terms $\frac{cn}{2}$

4 terms $\frac{cn}{4}$

$\longrightarrow$ total $= cn$

8 terms $T\left(\frac{n}{8}\right)$

$\vdots$

2^r terms $T\left(\frac{n}{2^r}\right)$ $T(1)$ . . . .

expand tree until $\frac{n}{2^r} = 1$, or $r = \log n$.

$T(n) =$ Sum of all the "$c.n$"-type terms in all the levels $+$ $T(1)$ terms in the last level

Sum of all the "$c.n$"-type terms in all the levels $\rightarrow$ $cn \cdot \log n$

$T(1)$ terms in the last level $\rightarrow$ $2^r \cdot T(1) = n \cdot 1$

$$T(n) = cn \log n + n.$$

$$= O(n \log n).$$

Tree: $T(n) = 2T(n/2) + cn$

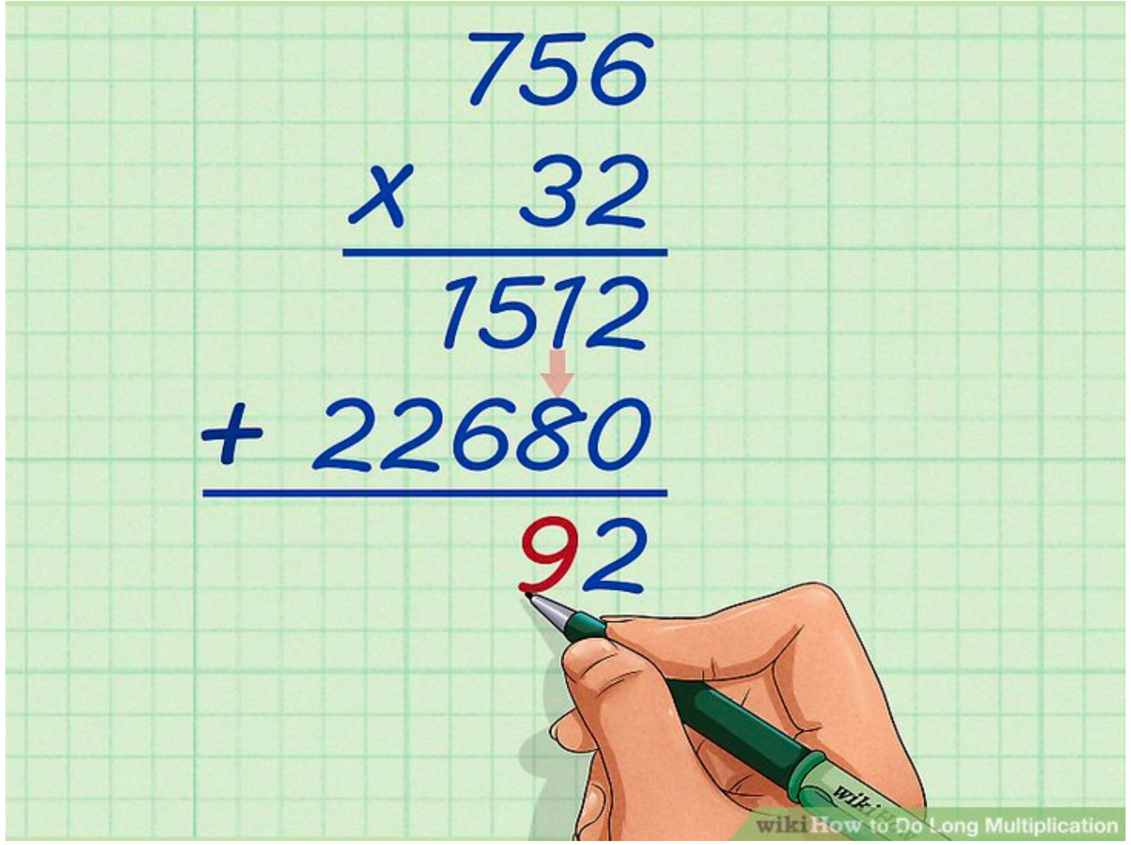
Plug-n-chug: $T(n) = 2T(n/2) + cn$

Guess-n-prove: $T(n) = 2T(n/2) + cn$

Example: integer multiplication

Problem: given two n-digit numbers $a = a_1a_2\ldots a_n$, and $b=b_1b_2\ldots b_n$, find the product a * b.

Elementary school algorithm



Divide and conquer?

Running time

Running time

Better algorithm?

Next class

- Complete multiplication example
- Linear time median
- Bring recurrence of choice — we'll try!