Recitation 1

Divide-and-Conquer

Rebecca Lin | Friday, September 6th, 2024

Divide-and-Conquer

1. **Divide** problem into subproblems of the same type
2. **Conquer** (solve) each subproblem recursively
3. **Combine** the solutions to a solution of the original problem

Example Recurrence:

"work to conquer subproblems" — $aT(\frac{n}{b})$

"work to reduce to and merge subproblems" — $f(n)$

$$T(n) = aT(\frac{n}{b}) + f(n)$$

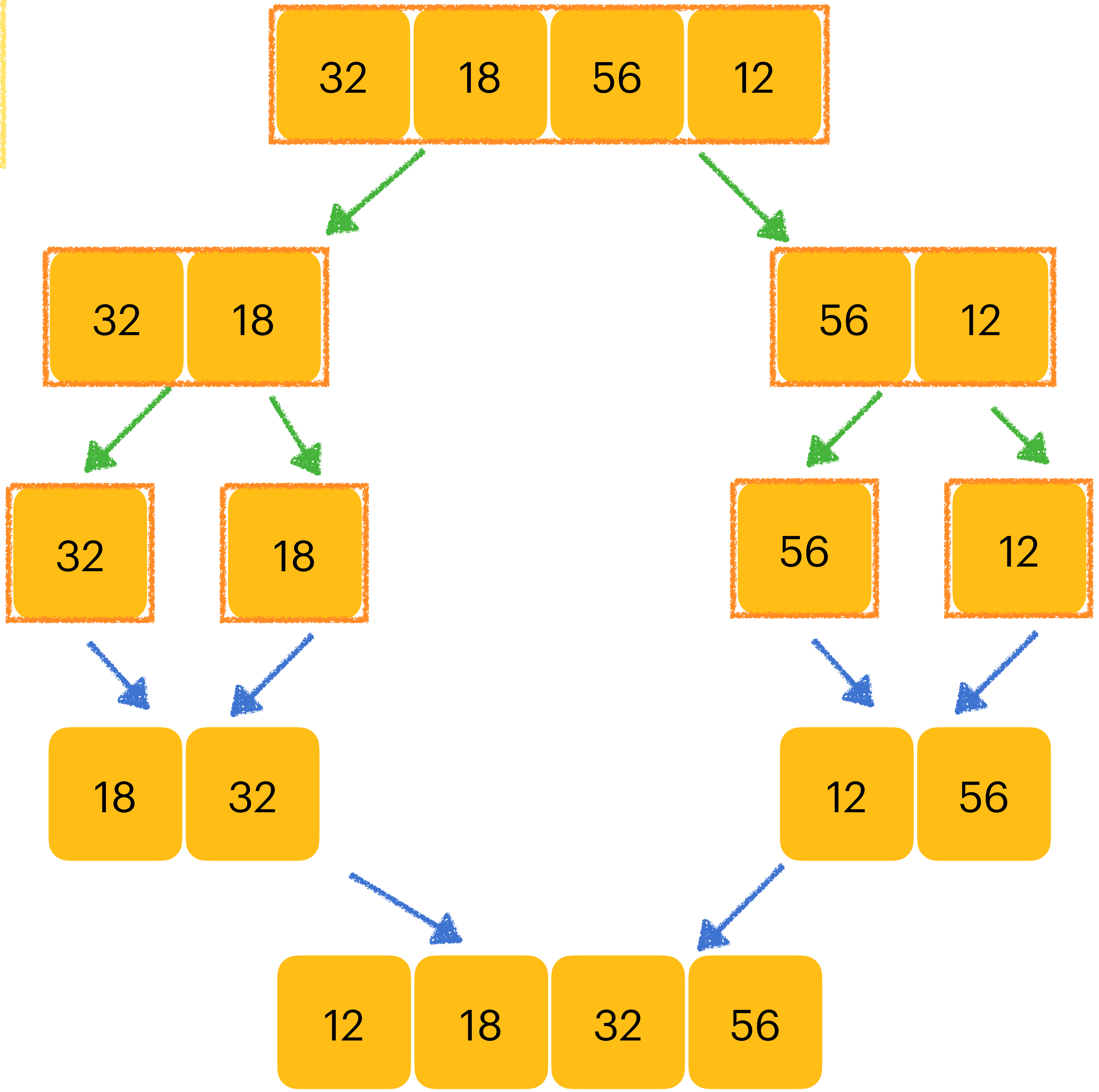
"split into $a \geq 1$ number of subproblems, each of size $\frac{n}{b}$ where $b > 1$"

Example: Merge Sort

$$T(n) = 2T(\frac{n}{2}) + O(n)$$

Base Case

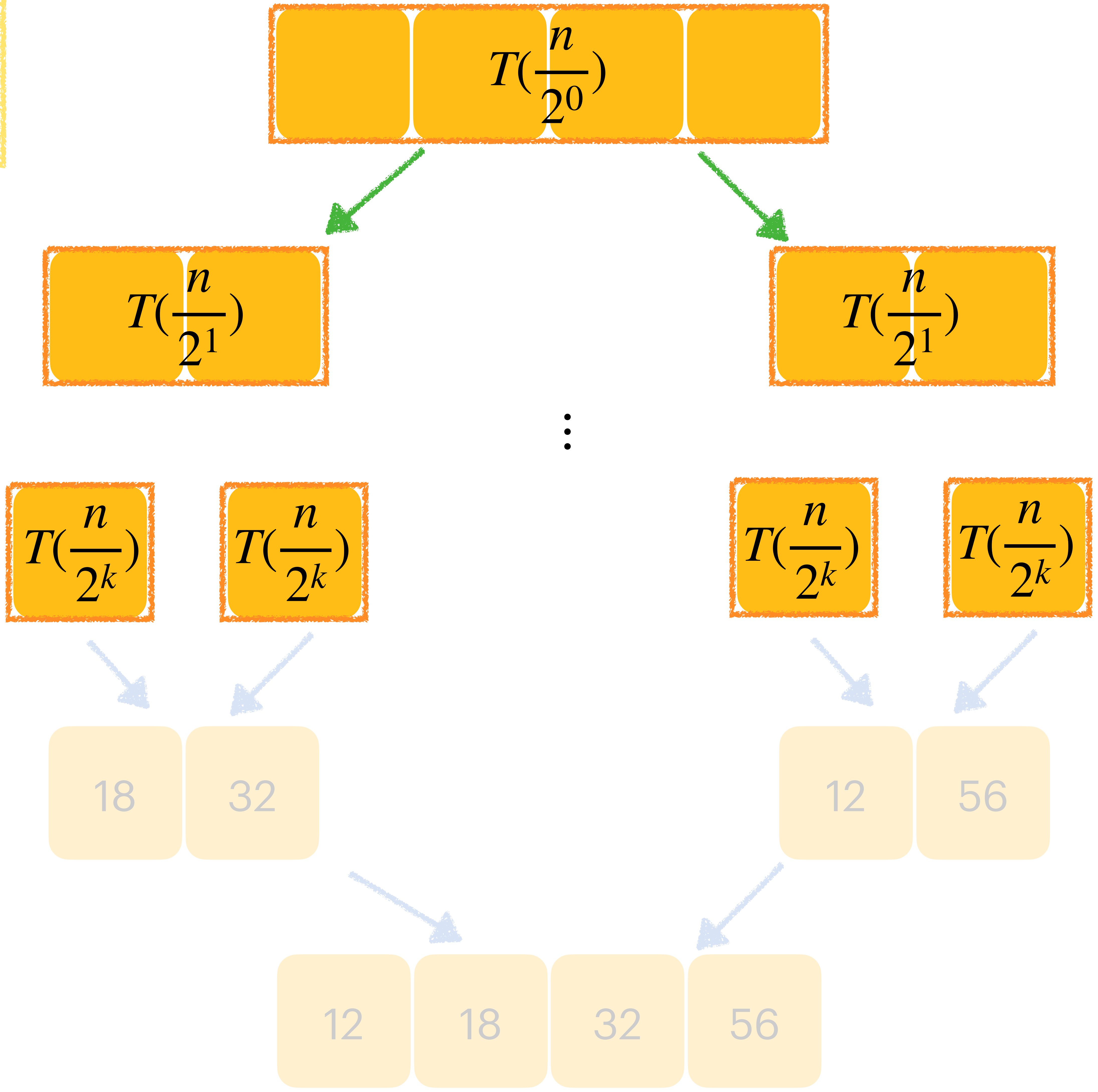
$T(1) = O(1)$



Example: Merge Sort

$$T(n) = 2T(\frac{n}{2}) + O(n)$$

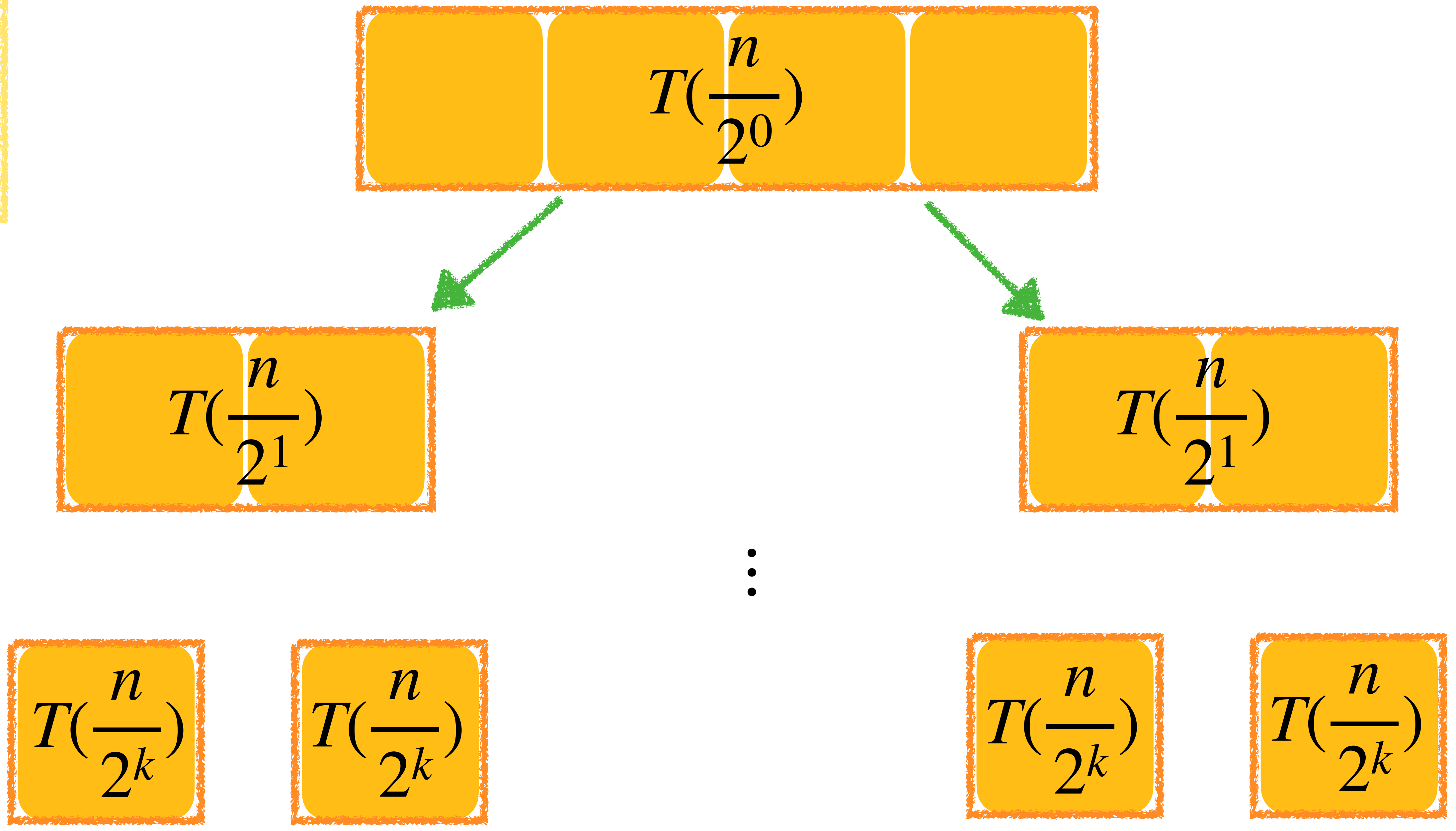
Base Case
$T(1) = O(1)$



$$O_0(n) + O_1(n) + \vdots + O_{k=\lg n}(n)$$

Example: Merge Sort

$$T(n) = 2T(\frac{n}{2}) + O(n)$$



Base Case
$T(1) = O(1)$

$$O_0(n) + O_1(n) + \vdots + O_{k=\lg n}(n)$$

$$O\left(\sum_{i=0}^{k} 2^i \cdot \frac{n}{2^i}\right) = O\left(\sum_{i=0}^{k} n\right) = O(kn) = O(n \log n)$$

Example: Maximum Subarray Sum

Given an array $A[1,\dots,n]$, find consecutive entries of A that yield the maximum sum.

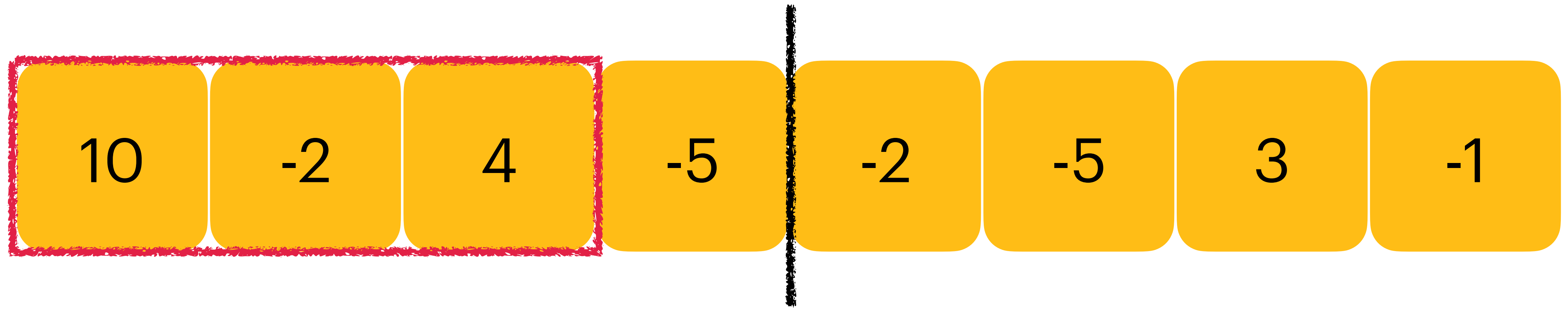
Example 1	12	7	8	2	15	9	24	32
Example 2	-2	-9	-4	-22	-7	-14	-21	-17
Example 3	1	-7	5	0	2	-1	3	-4

What are some naive solutions?

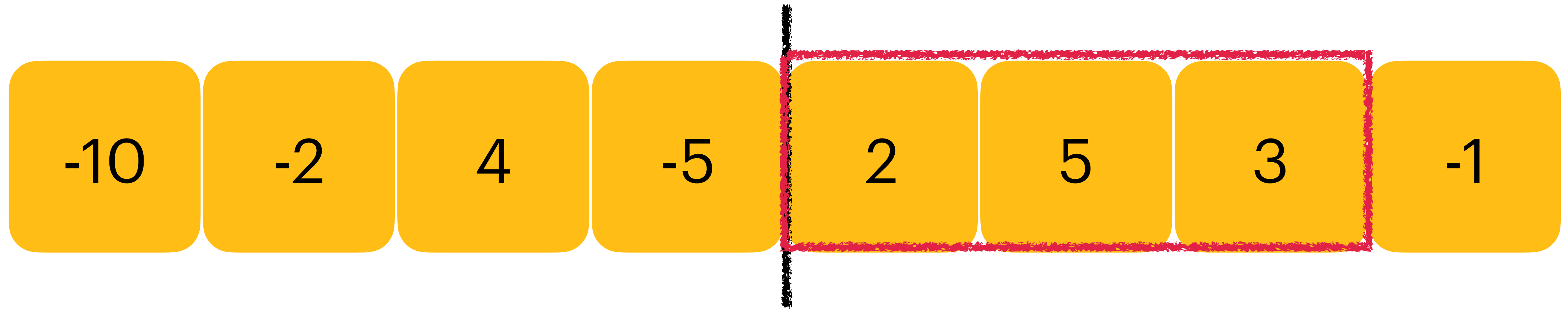
Example: Maximum Subarray Sum

Divide-and-Conquer

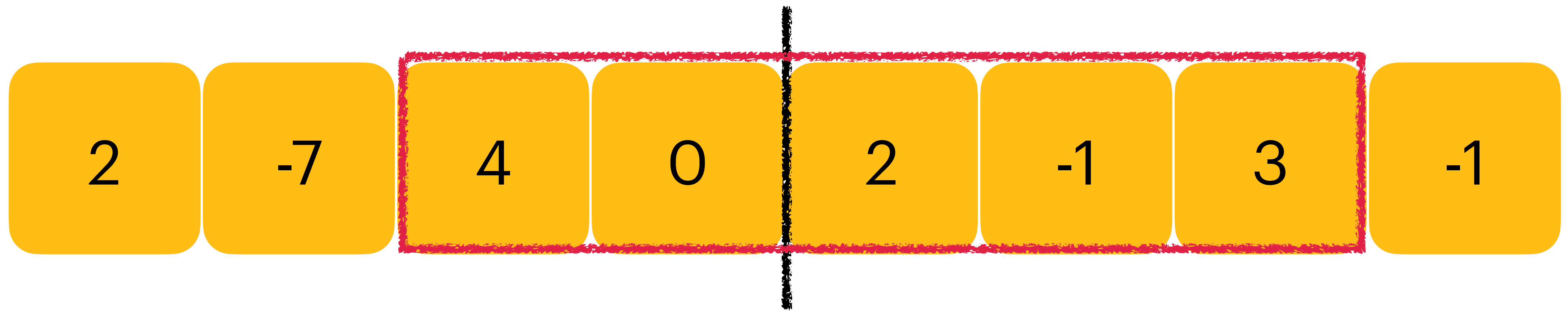
Case 1: Best solution entirely in left subarray



Case 2: Best solution entirely in right subarray



Case 3: Best solution crosses partition



Example: Maximum Subarray Sum

Divide-and-Conquer

1. Recurse:
   - $maxL \leftarrow$ best solution left of partition
   - $maxR \leftarrow$ best solution right of partition
2. Compute best solution $maxM$ crossing partition
3. Return the best of $maxL, maxR, maxM$

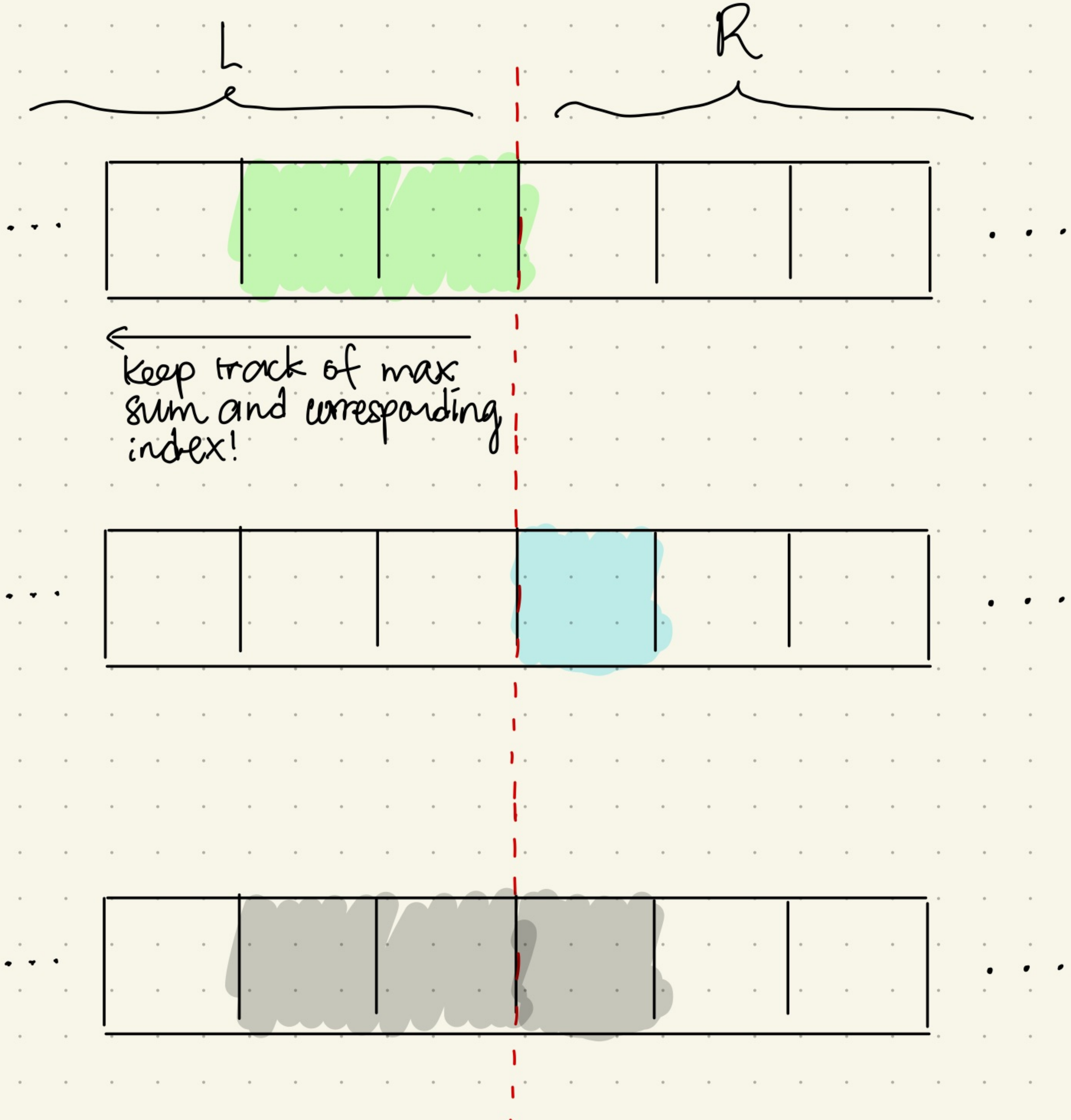
Claim: Step 2 — computing $maxM$ — can be performed in $\Theta(n)$ time. *How?*

Compute maxM in $\Theta(n)$ Time

1. Find max-sum subsequence in L right-aligned at the partition.

2. Find max-sum subsequence in R left-aligned at the partition.

3. Return the union of the two max-sum sequences.



Techniques for Solving Recurrences

Recurrence tree

- Slides: Merge Sort
- Recitation Handout: Section 3.5: Practice Problems, e.g., Q4b; Appendix A

Unrolling

- Recitation Handout: Section 3.1, e.g., $T(n) = 2T(\frac{n}{2}) + \Theta(\frac{n}{\log n})$; Appendix B

Substitution

- Recitation Handout: Section 3.2; Section 3.5, e.g., Q3

Master Theorem

- Recitation Handout: Section 3.1 (note when it *cannot* be used); Section 3.5, e.g., Q1, Q2

Guess-and-check

- Recitation Handout: Section 3.3 (note common mistakes); Section 3.5, e.g., Q4a, Q5, Q6, Q7

Master Theorem

Recurrence: $T(n) = aT(\frac{n}{b}) + f(n)$, where $a \geq 1, b > 1$

Three cases depending on the value of $n^{\log_b a}$:

1. If $f(n)$ is polynomially less than $n^{\log_b a}$ (i.e., $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$), then $T(n) = \Theta(n^{\log_b a})$
2. If $f(n) = \Theta(n^{\log_b a} \log^k n)$ for some constant $k \geq 0$, then $T(n) = \Theta(f(n) \log n)$
3. If $f(n)$ is polynomially greater than $n^{\log_b a}$ (i.e., $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$), then $T(n) = \Theta(f(n))$

Idea: Compare weight at the root of recurrence tree, $f(n)$, to the # of leaves, $n^{\log_b a}$

Substitution: Worked Example (Section 3.2)

uh oh!

$T(n) = 2T(\sqrt{n}) + 1.$

- Define m s.t. $n = 2^m$

- Substitute: $T(2^m) = 2T(2^{\frac{m}{2}}) + 1$

- Create new function $S(m) = T(2^m)$

  $\Rightarrow \quad S(m) = 2S(\frac{m}{2}) + 1 = O(m)$

  Master Theorem

  $a = 2, b = 2$, so $n^{\log_b a} = n$

  $f(n) = 1.$

  $\Rightarrow$ Case 1.

- Resubstitute: $T(2^m) = O(m)$

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

  $m = \lg n$

Other Notes

- Review recitation notes:
  - *Lots* of practice problems, e.g., Section 3.5, Appendix AB, etc.
  - Recap of lecture (Section 4)
  - Asymptotic Notation Reference (Section 5)
- Watch out for my Canvas note:
  - Link to recitation slides
  - Form for anonymous feedback
- Probability review this Sunday 9/8
- Email: ryelin@mit.edu