Johannes Fischer

Advanced Data Structures

MSc-Vorlesung
Wintersemester 2011/12
KIT

Preliminaries

- 5 ECTS
- lectures in German, slides etc. in English
- prequisites:
  - Algorithmen II
  - interest in discrete, combinatorial problems
- 15 lectures (NOT on December 8th)
- oral exam (20 mins)

Preliminaries

- course homepage:
  `http://algo2.iti.kit.edu/1909.php`
  - slides (& script)
  - hints for LaTeX and mathematical writing
- Johannes.Fischer@kit.edu (room 206)
- office hours: Monday 14-15 (NOT Dec. 5th)

No „Übungen“! Scribing!

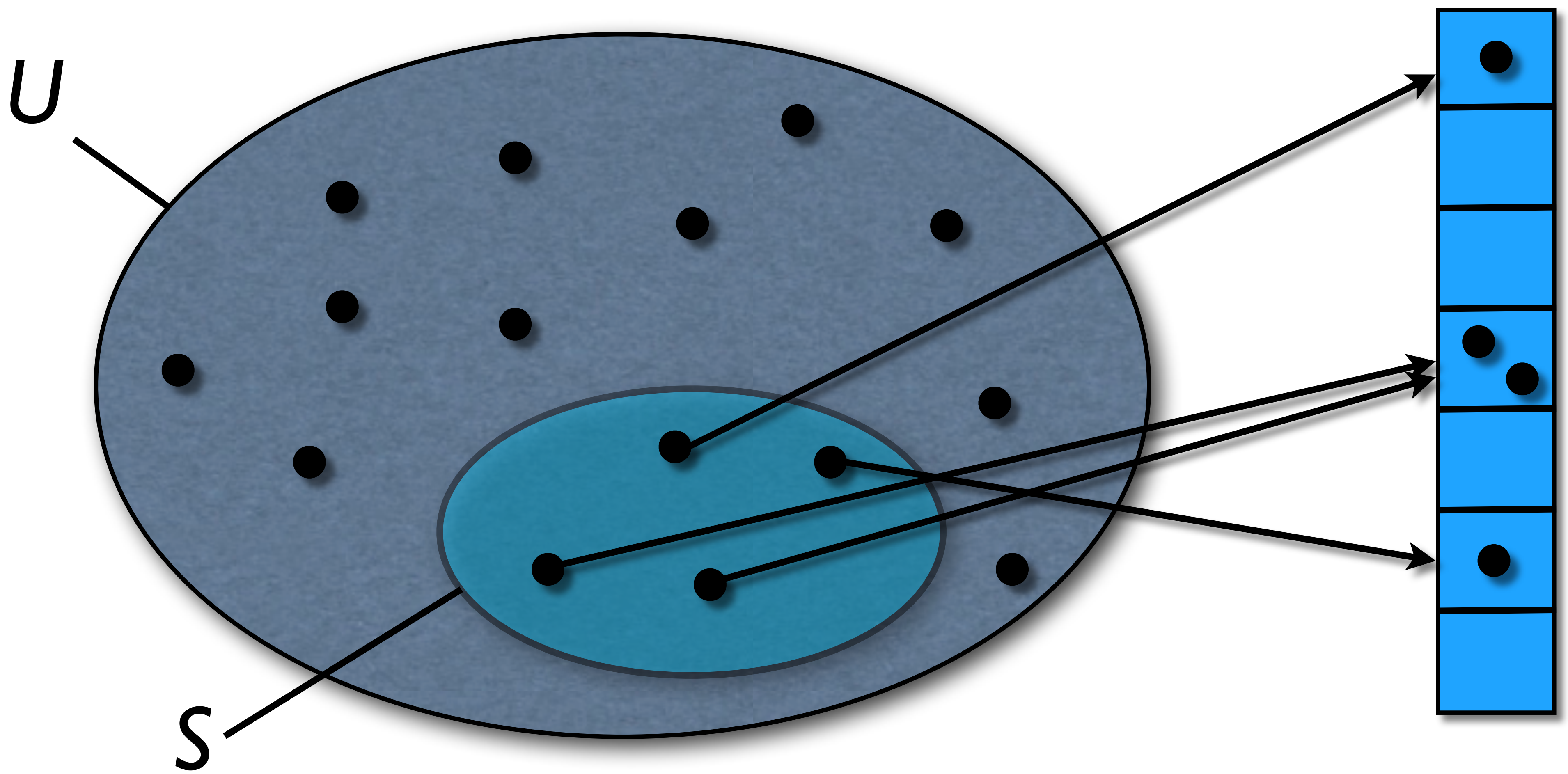
- write script for one lecture
- material:
  - slides
  - my notes
  - research literature (NOT wikipedia etc.)
- use LaTeX $\Rightarrow$ learn to write scientifically
- vector graphics: ipe, xfig, ...

Course Contents

- hashing
- predecessor data structures
- integer sorting/ searching
- distance oracles
- tree labelings
- lowest common/ level ancestors
- range (minimum) queries
- succinct trees
- text indexing (string B-trees)

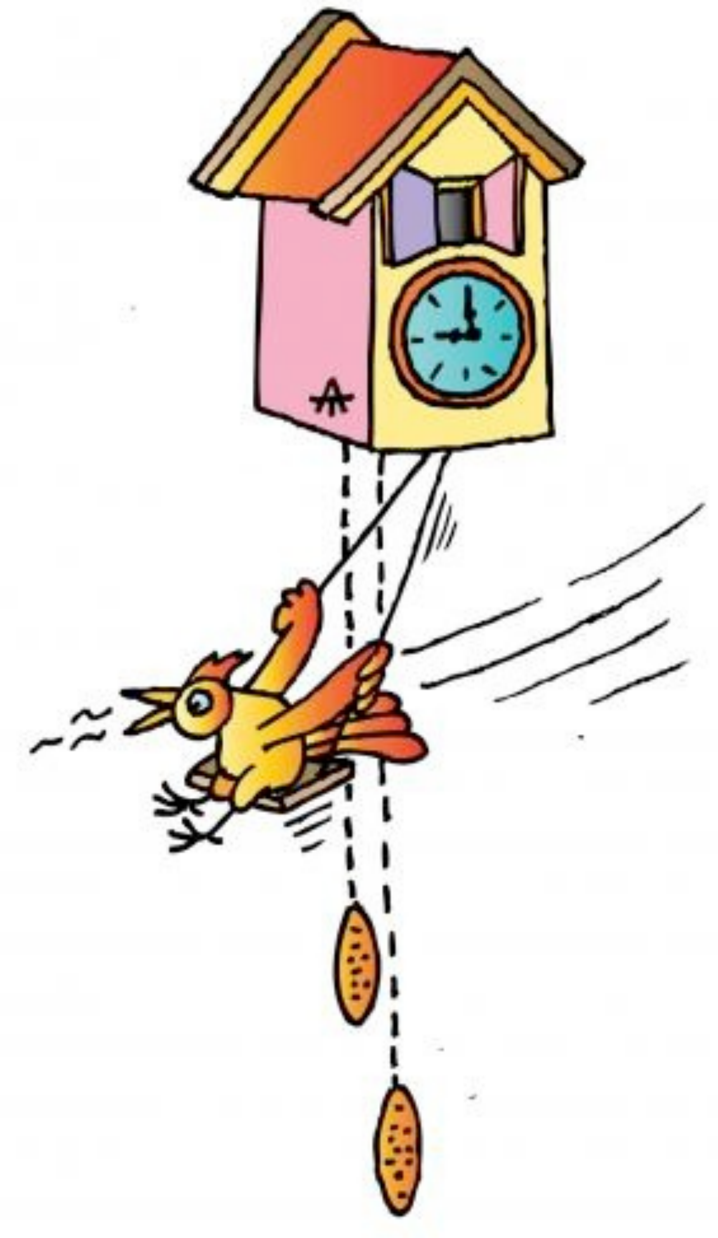
Hashing

- set S of n objects from a LARGE universe U
- query for membership (+satellite info)
- Use space $O(n)$, not $O(|U|)$



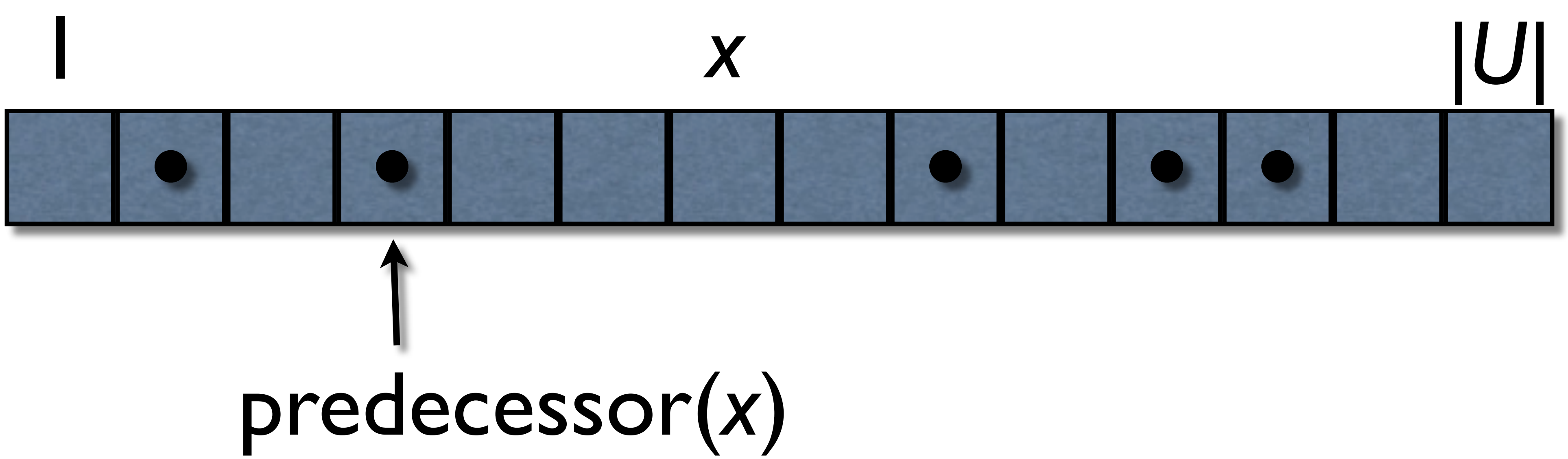
Hashing: lookup time

- chaining/linear probing:
  $O(1)$ **expected** time
- cuckoo hashing:
  $O(1)$ **worst case** time
- other operations $O(1)$
  amortized & expected



Predecessor Queries

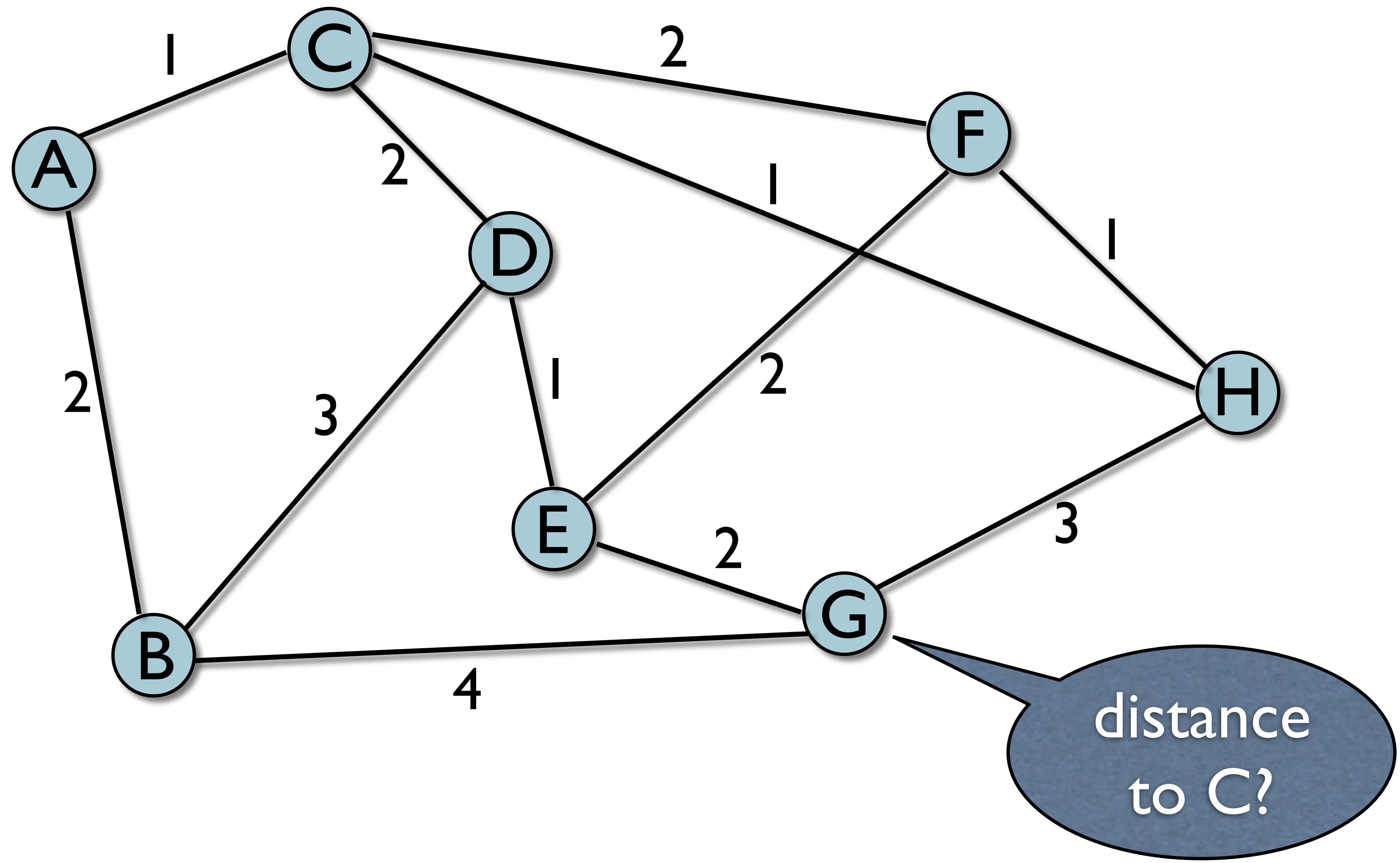
- *S*: *n* objects from a SORTED universe *U*
- given $x \in U$, return $\max\{y \leq x : y \in S\}$
- fast if elements are integers: $O(\lg\lg |U|)$



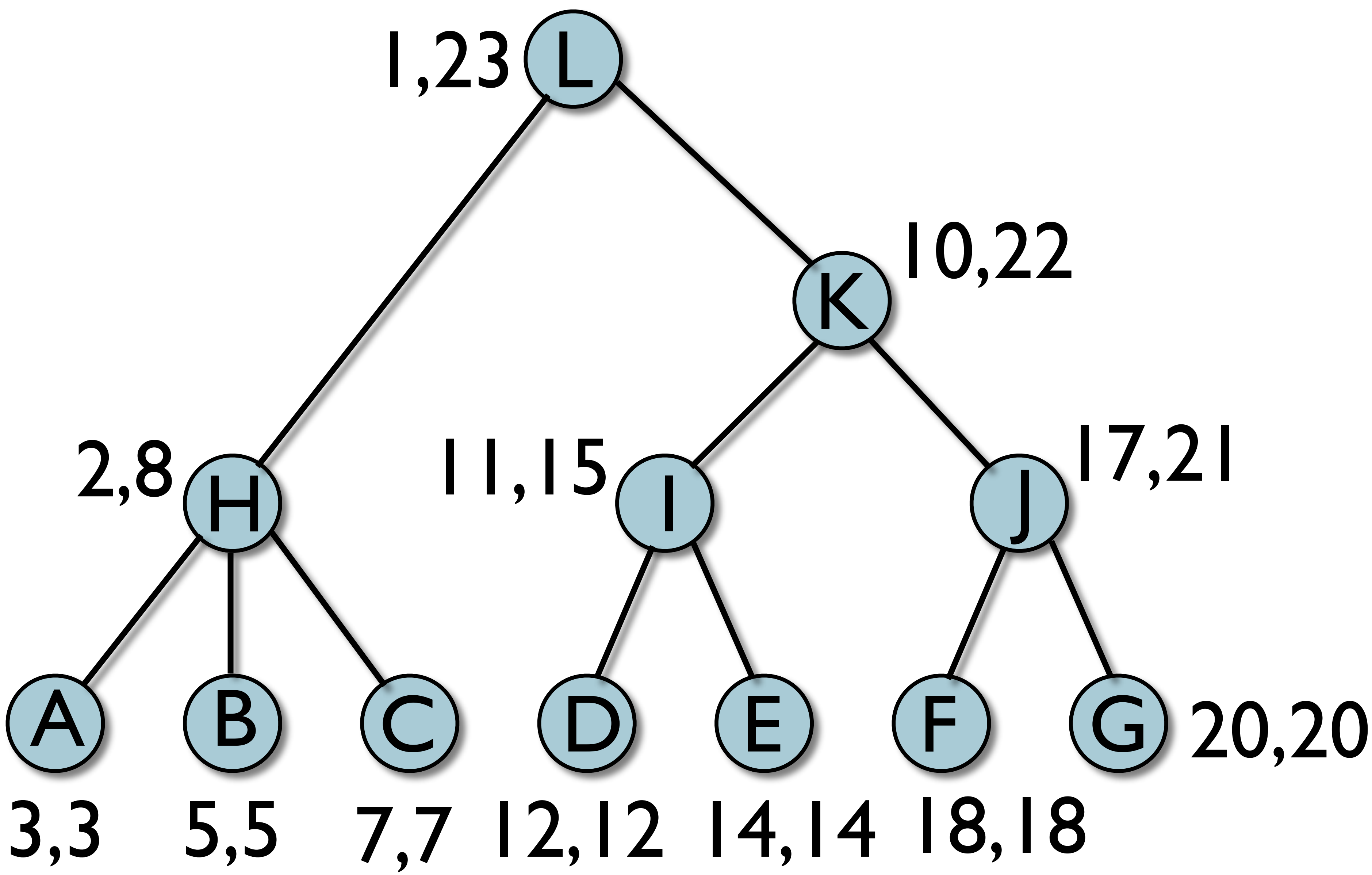
Integer Sorting

- sort n elements from a universe $[0,2^w-1]$
- comparison based sorting: $\Theta(n \lg n)$
- counting sort: $O(n + 2^w)$
- with predecessor queries: $O(n \lg w)$
- **signature sort**:
  - $O(n)$ for w sufficiently large
  - $O(n \lg\lg n)$ for all w

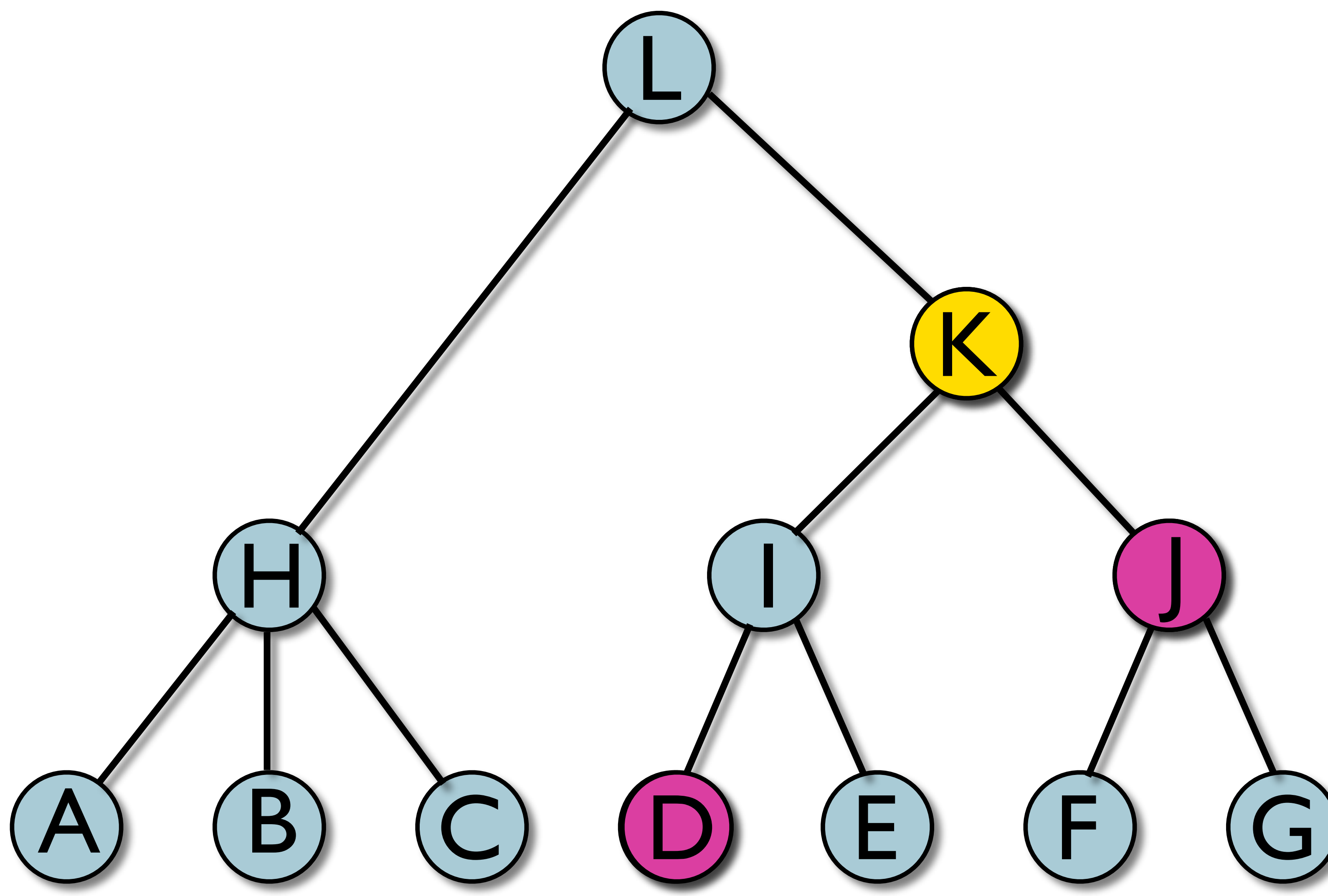
Distance Oracles



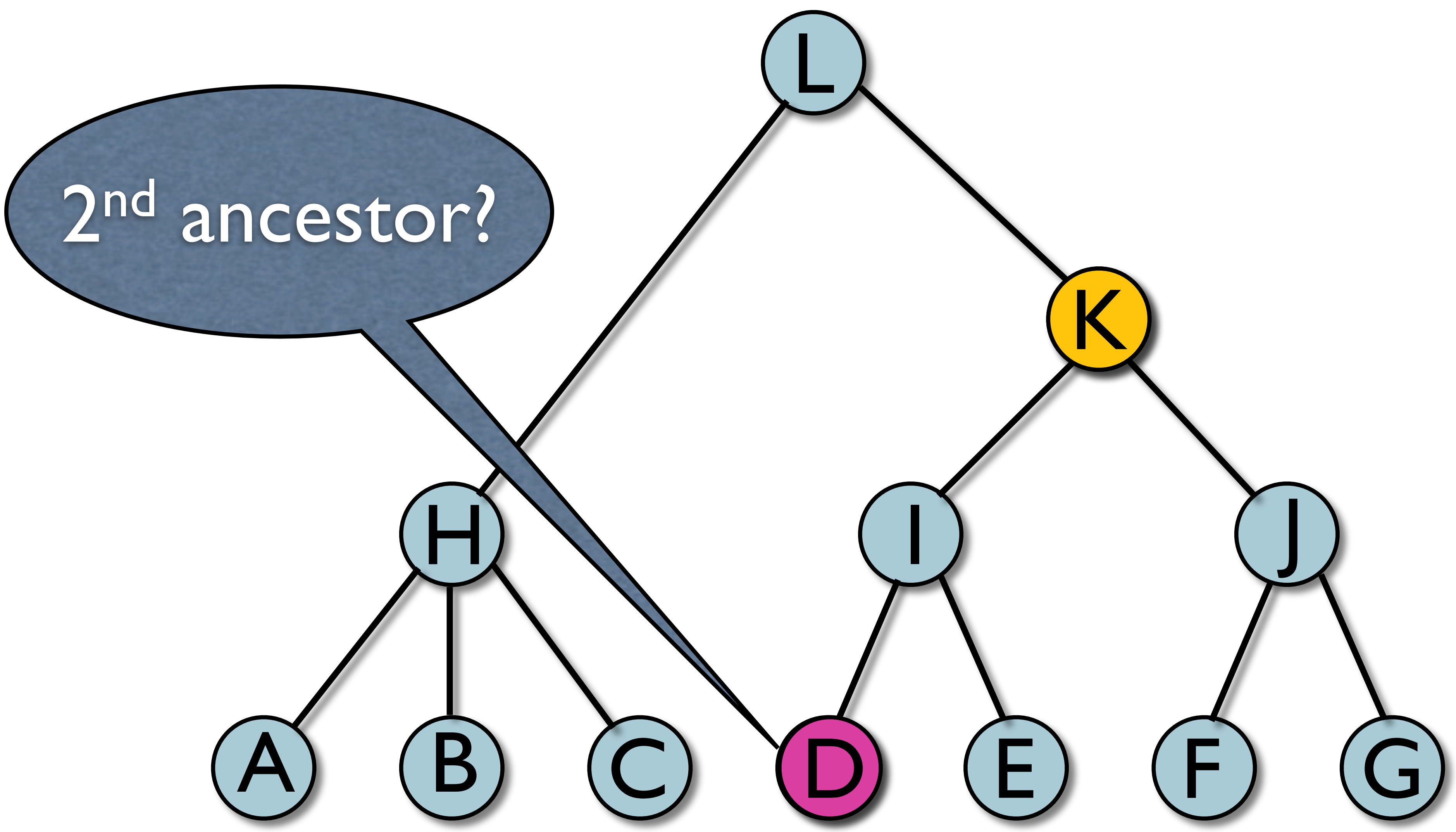
Tree Labelings: Ancestors



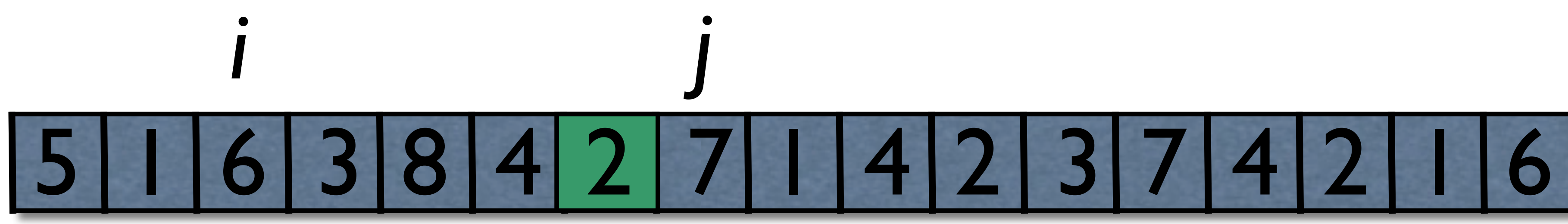
Lowest Common Ancestors



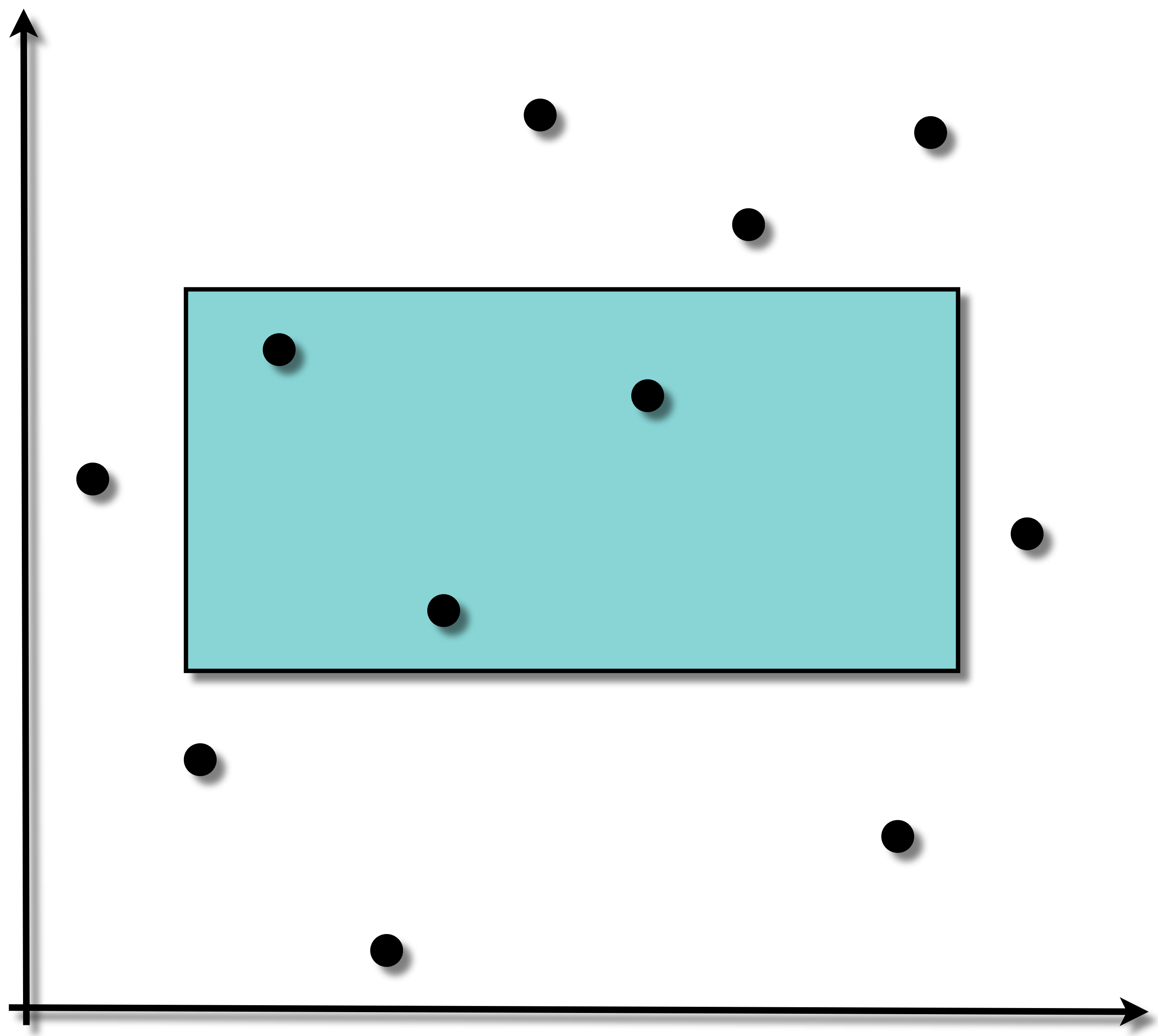
Level Ancestors



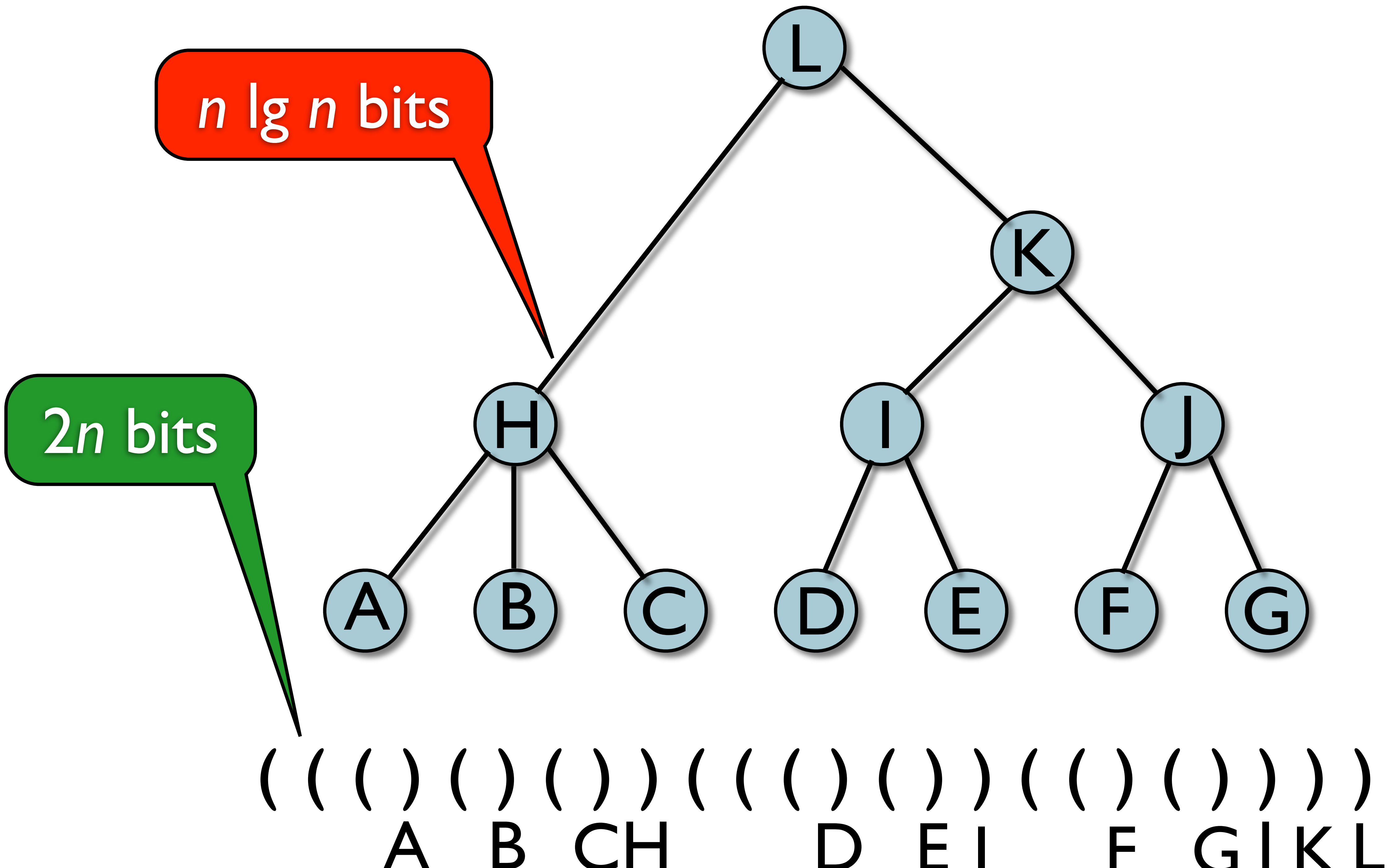
Range Minimum Queries



2d Range Reporting

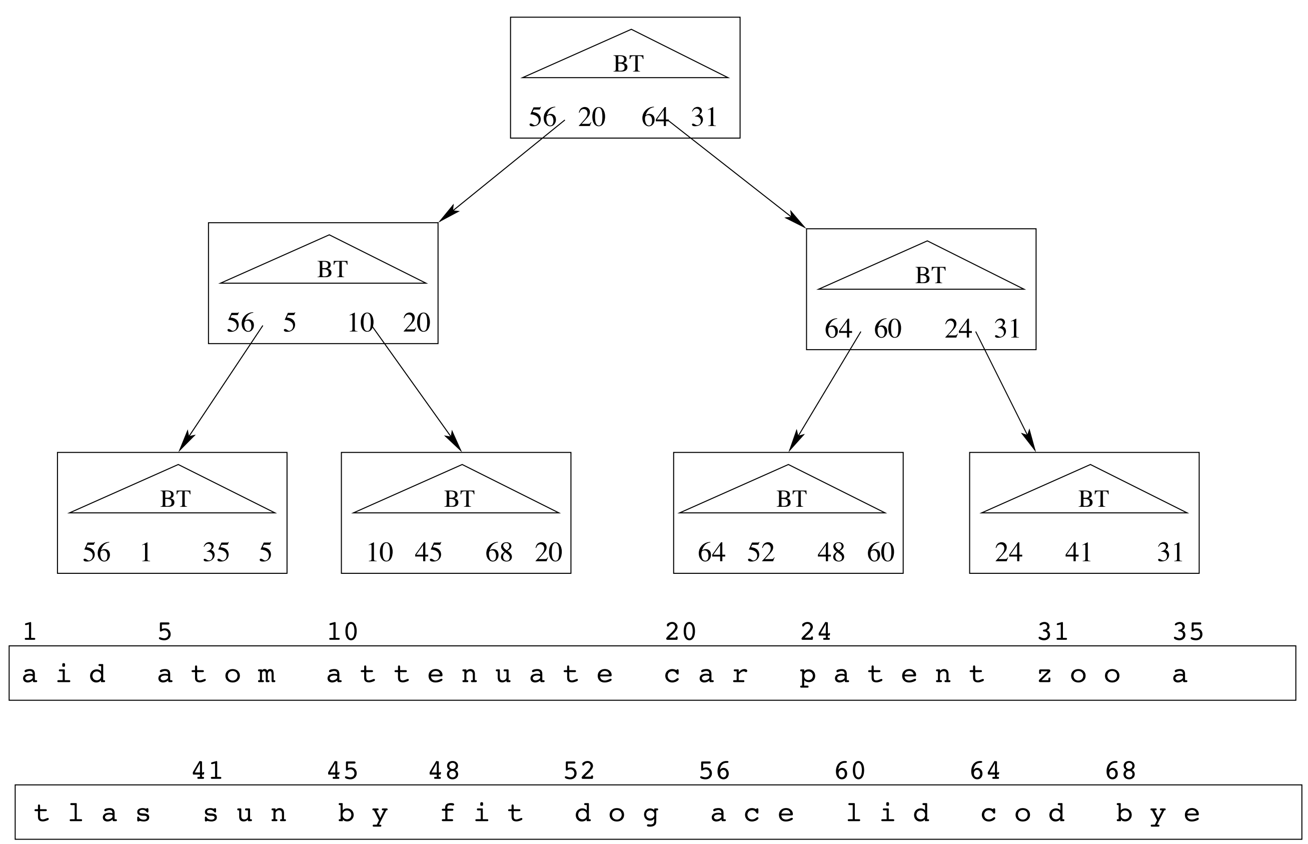


Succinct Trees



String B-Trees

- text indexing in **external** memory
- substring queries (cf suffix tree/array)
- new challanges (minimize IOs)



Theory vs. Practice

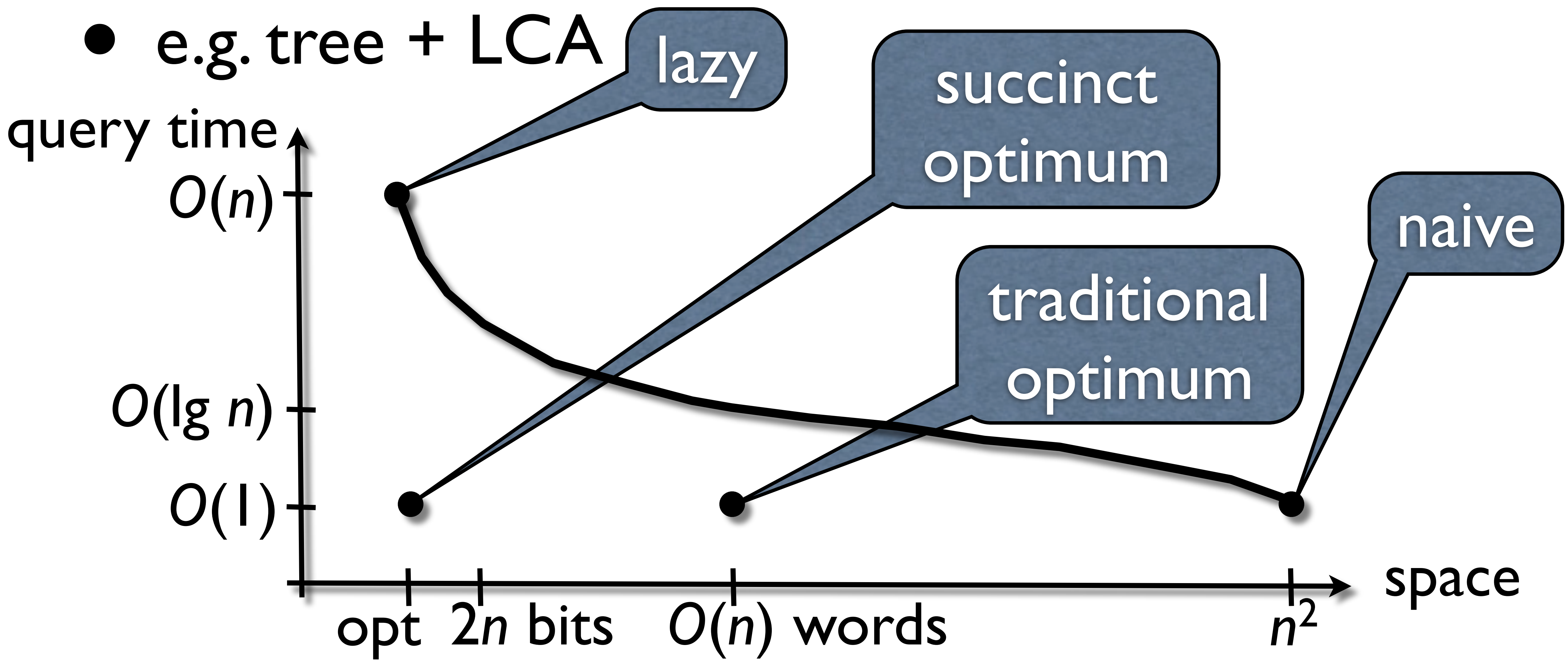
- focus on **theoretical** (=mathematical) analysis of data structures
- BUT: most methods highly **practical** (perhaps with some engineering effort)
  - VL "Algorithm Engineering"
- every method better than naive approach (complex analysis $\not\Rightarrow$ slow running time)

Classification of DSs

object	type of DS
numbers	„normal“
point sets	integer
graphs	randomized
trees	distributed
arrays	succinct
strings	external
...	parallel
	cache aware etc.

What is a DS?

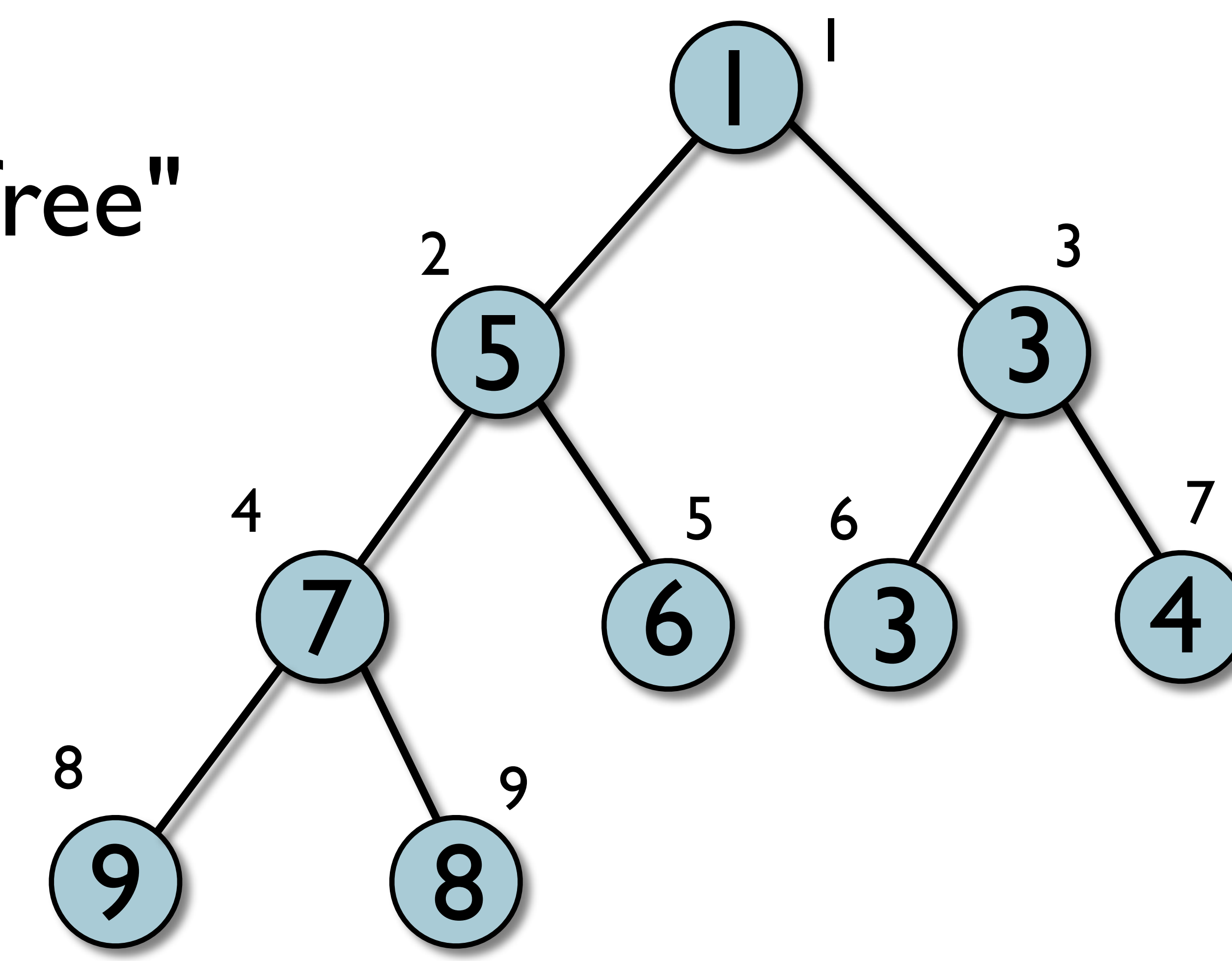
- extend functionality
  - ADT + DS = ADT' with ADT' ⊇ ADT
- **tradeoff** time/space
- e.g. tree + LCA



Implicit DS

- clever storage
  - functionality "for free"
- e.g. heap:

$$\text{parent}(x) = \left\lfloor \frac{x}{2} \right\rfloor$$



1	2	3	4	5	6	7	8	9
1	5	3	7	6	3	4	9	8