MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

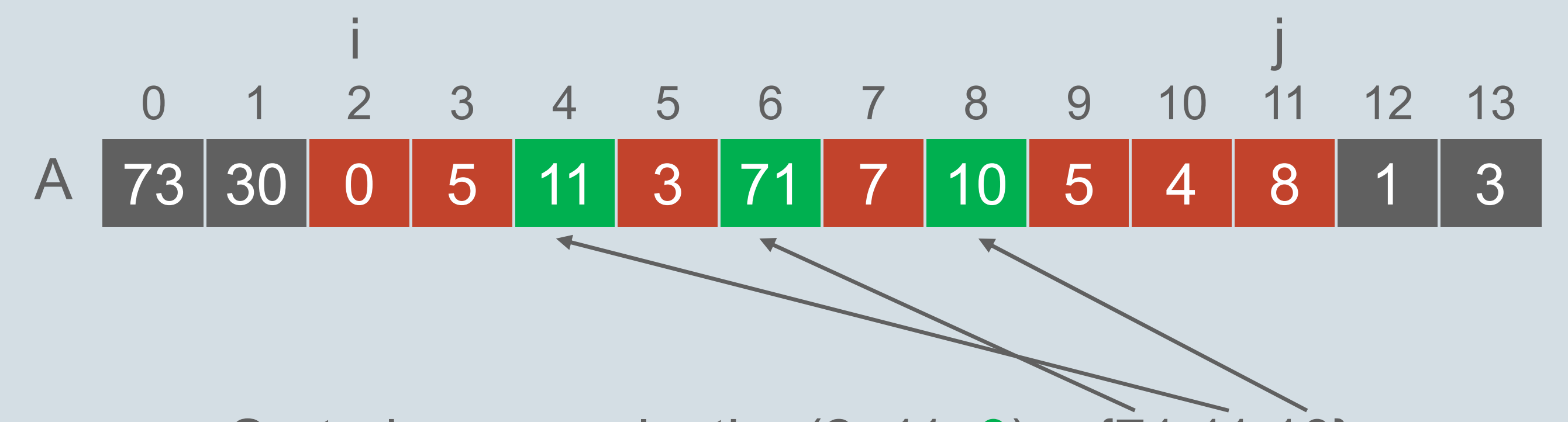
Mark Greve
Aarhus University

Online Sorted Range Reporting

Problem

Preprocess an array A of n elements into a space efficient data structure supporting the following queries.

Sorted range selection(i, j, k)
Report the k largest elements in A[i..j] in **sorted** order.



Sorted range selection(2, 11, 3) = {71,11,10}

Online sorted range reporting(i, j)
Report the elements in A[i..j] one-by-one in **sorted** order.

Results (RAM)

Space usage	O(n) words
Preprocessing	O(n log n) time
Sorted range selection	O(k) time
Online sorted range reporting	O(1) time per element

References

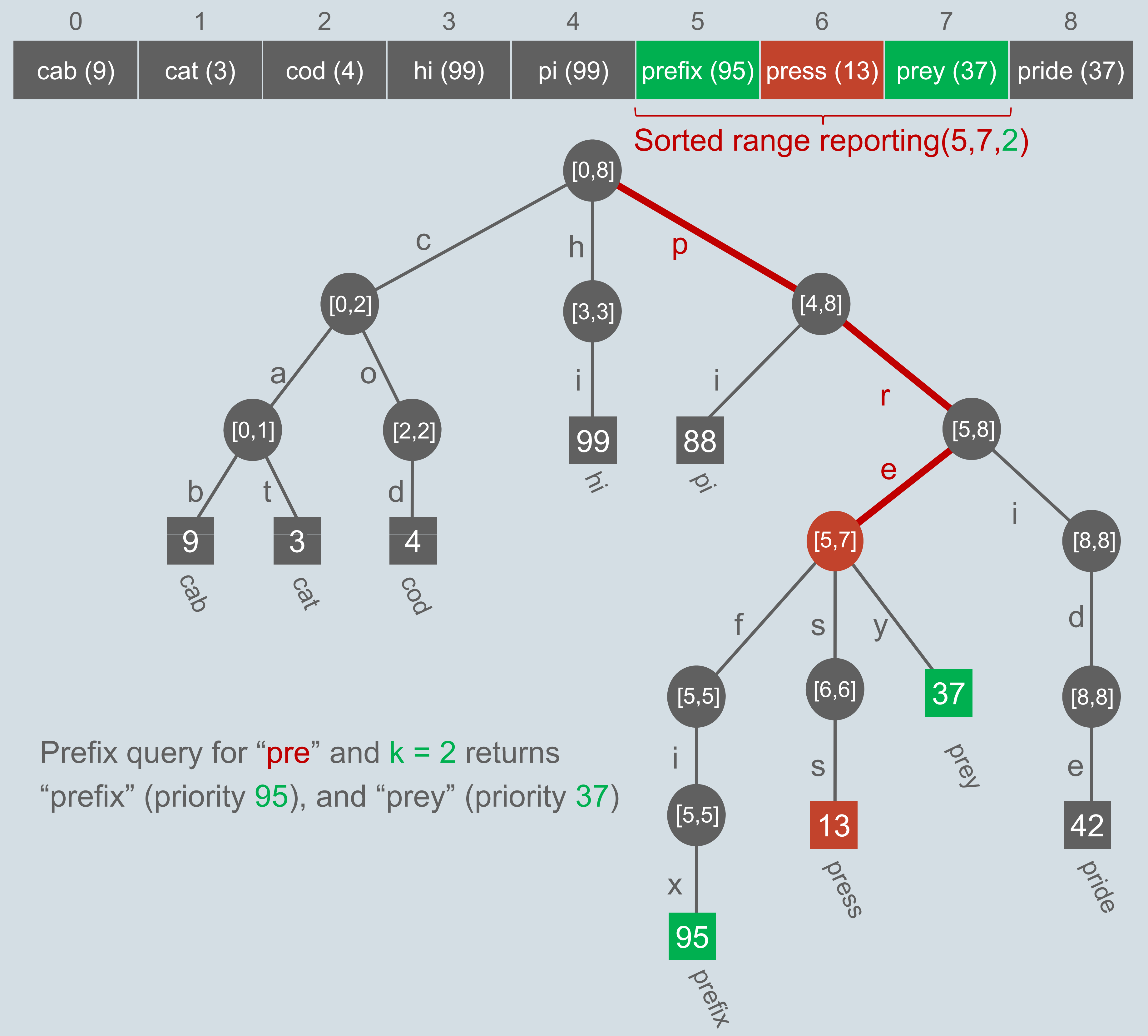
[1] G. S. Brodal, R. Fagerberg, M. Greve, and A. López-Ortiz, *Online Sorted Range Reporting*. In submission.

[2] G. Frederickson and D. B.Johnson, *The Complexity of Selection and Ranking in X+Y and Matrices with Sorted Columns*, Journal of Computer and System Sciences 24(2): 197-208 (1982).

Applications

Prioritized string prefix queries

- Input : A set of strings with associated priorities.
- Queries : Report the k strings of highest priority with a given prefix.
- Solution : Trie + sorted range reporting.



Prefix query for “pre” and k = 2 returns “prefix” (priority 95), and “prey” (priority 37)

Other applications

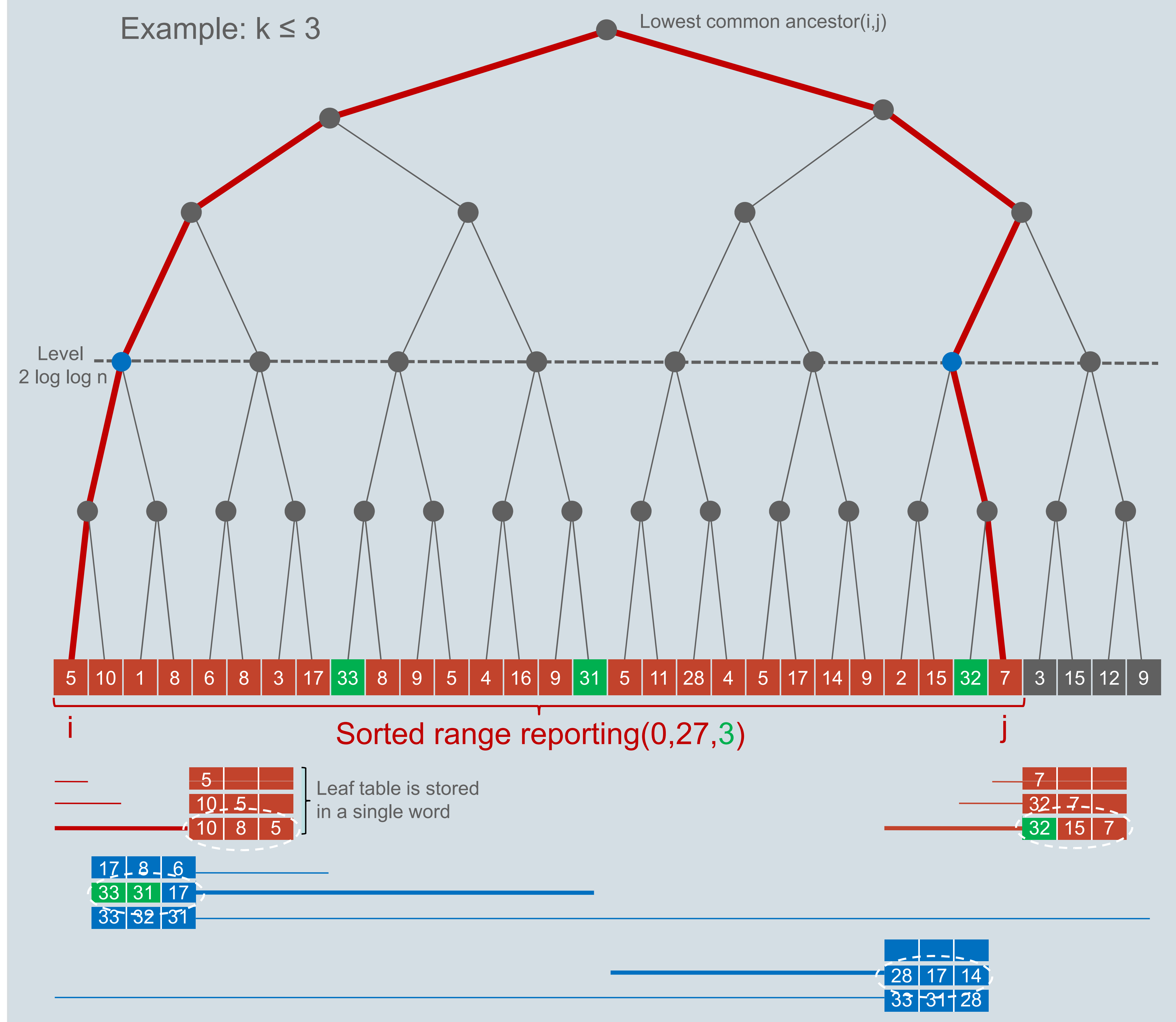
Candidate problems are those whose solution involves sorting to compute intersections of sets. Using our data structure the output is returned in **sorted** order, and so the sorting step will be unnecessary. Such problems can be found in areas such as search engines for information retrieval and databases.

Main Ideas in the Solution

Maintain O(log log n) structures, one structure for $k \le \log n/(2 \log\log n)^2$ using space O(n log n) bits and one structure for each i where $2^{2^i} \le k < 2^{2^{i+1}}$ using space $O(n \cdot 2^i)$ bits. In total space O(n log n) bits.

Case $k \le \log n/(2 \log\log n)^2$

For leaves and nodes at level 2 log log n, precompute answers for all possible queries and store compactly. To answer a query we locate the lowest common ancestor of the leaves, then lookup ≤ 4 precomputed sorted sets from which we extract the k overall largest elements by merging.



Case $k > \log n/(2 \log\log n)^2$

Cut the i’th tree into three parts by cutting at levels 2^i and $7 \cdot 2^i$. Queries are handled by combining O(1) queries from each of the parts. Top part is handled as above. Queries in the middle and bottom parts use Frederickson's array selection algorithm [2], radix sorting, and compact representation of precomputed sorted sets.

MADALGO – Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation