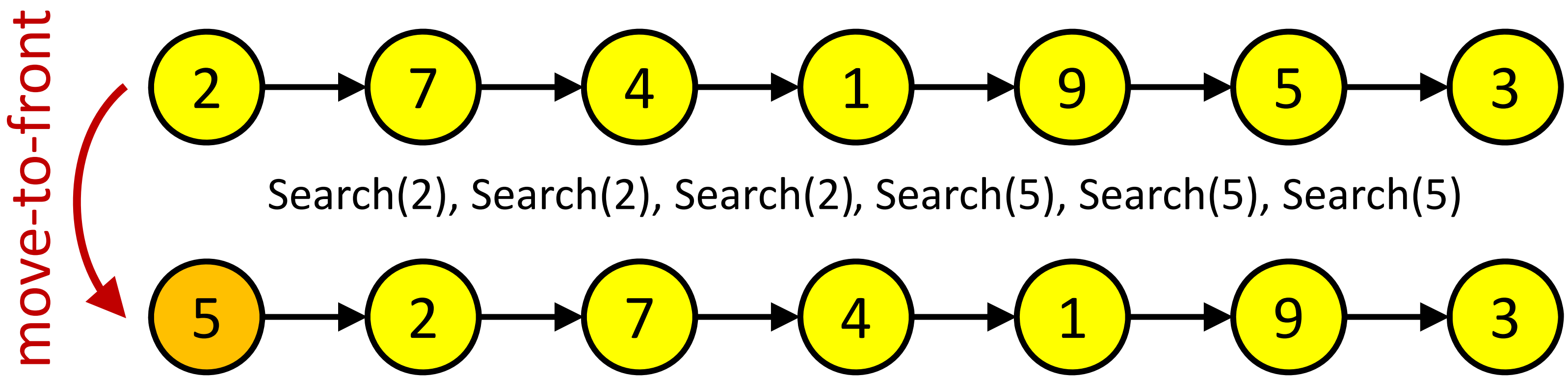
Unified Access Bound

[M. Bădoiu, R. Cole, E.D. Demaine, J. Iacono, *A unified access bound on comparison-based dynamic dictionaries*, Theoretical Computer Science, 382(2), 86-96, 2007]

- Dictionary: Insert(x), Delete(x), Search(x)
- Comparison model

Solution 1: Balanced search tree ⟶ O(log n)

Solution 2: Unordered linked list ⟶ O(n)

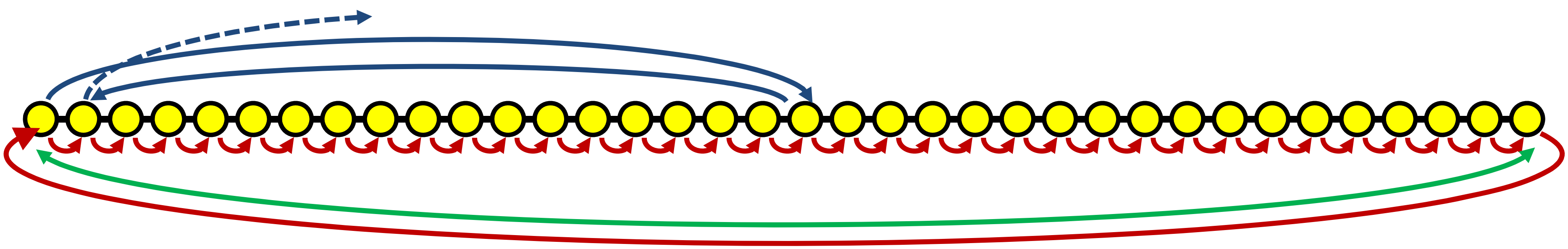


[D.D. Sleator, R.E. Tarjan, *Amortized Efficiency of List Update Rules*, Pro. 16th Annual ACM Symposium on Theory of Computing, 488-492, 1984]

The paper initiated the study of competitivenes analysis of **online algorithms** for list ordering, search-trees, paging algorithms, ... (move-to-front is **2-competitive**)

Access sequences - examples

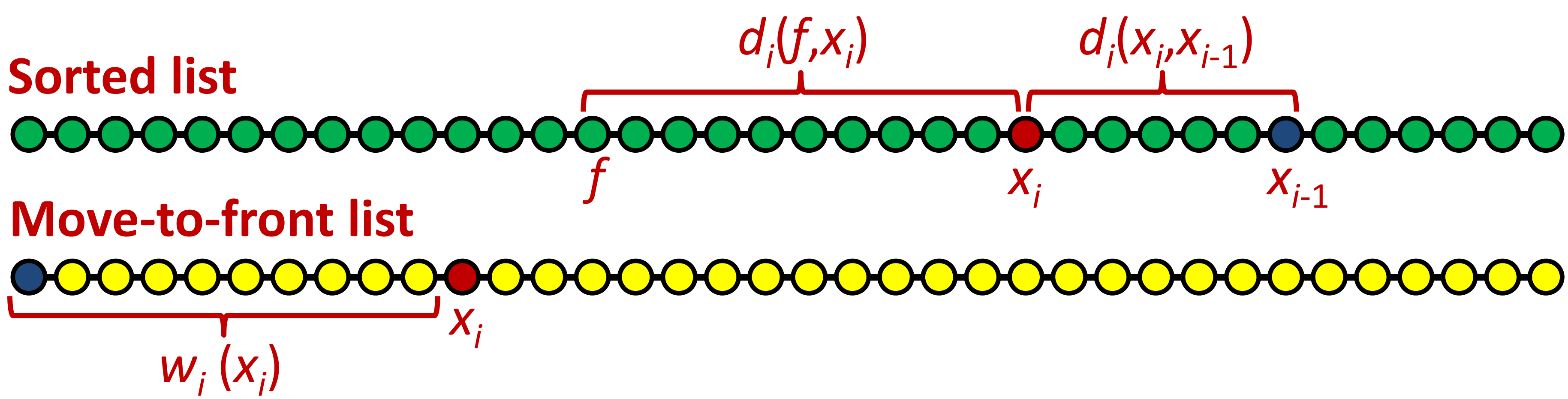
- $X_1 = 1, 2, 3, \ldots, n, 1, 2, 3, \ldots, n, 1, 2, 3, \ldots$
- $X_2 = 1, n, 1, n, 1, n, \ldots$
- $X_3 = 1, n/2, 2, n/2+1, 3, n/2+2, \ldots, n/2, n, 1, \ldots$



Access sequence $X = (x_1, x_2, \ldots, x_m)$

- Static optimal $O(\log (1/p(x_i)))$
- Sequential-access bound $O(1)$
- Static finger bound $O(\log d_i(f,x_i))$
- Dynamic finger bound $O(\log d_i(x_i, x_{i-1}))$
- Working set bound $O(\log w_i(x_i))$
- Unified bound $O(\min_{y \in S_i} \log(w_i(y)+d_i(x_i, y)))$

Finger search tree



Splay trees (amortized)

- [1] Static optimal $O(\log(1/p(x_i)))$
- [2] Sequential-access bound $O(1)$
- [1] Static finger bound $O(\log d_i(f,x_i))$
- [3] Dynamic finger bound $O(\log d_i(x_i, x_{i-1}))$
- [1] Working set bound $O(\log w_i(x_i))$
- Open Unified bound $O(\min_{y \in S_i} w_i(y)+d_i(x_i, y))$

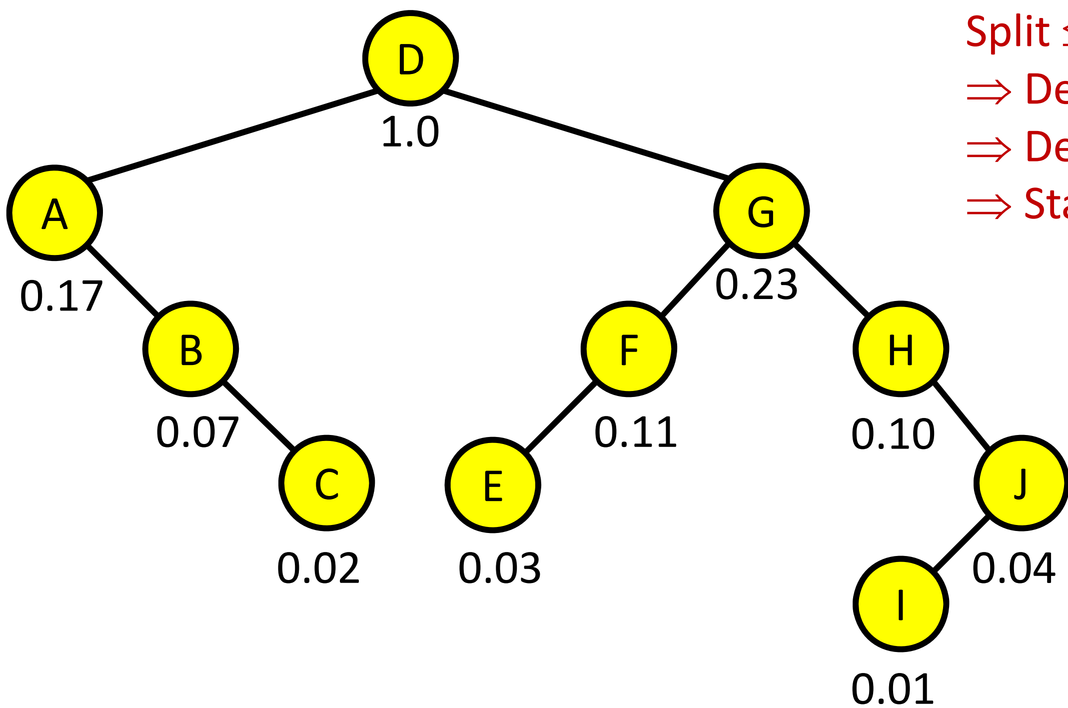
1 [D.D. Sleator, R.E. Tarjan, *Self-Adjusting Binary Search Trees*, J. ACM 32(3), 652-686, 1985]

2 [R.E. Tarjan, *Sequential access in play trees takes linear time*.
Combinatorica 5(4), 367-378, 1985]

3 [R. Cole, B. Mishra, J.P. Schmidt, A. Siegel, *On the Dynamic Finger Conjecture for Splay Trees*.
Part I: Splay Sorting log n-Block Sequences. SIAM J. Computing, 30(1), 1-43, 2000]
[R. Cole, *On the Dynamic Finger Conjecture for Splay Trees. Part II: The Proof*.
SIAM J. Computing, 30(1), 44-85, 2000]

Static optimality

[T.C. Hu, A.C. Tucker, *Optimal computer search trees and variable-length alphabetic codes*, SIAM Journal on Applied Mathematics 21 (4), 514–532, 1971]
[D.E. Knuth, *Optimum binary search trees*, Acta Informatica 1, 14–25, 1971]



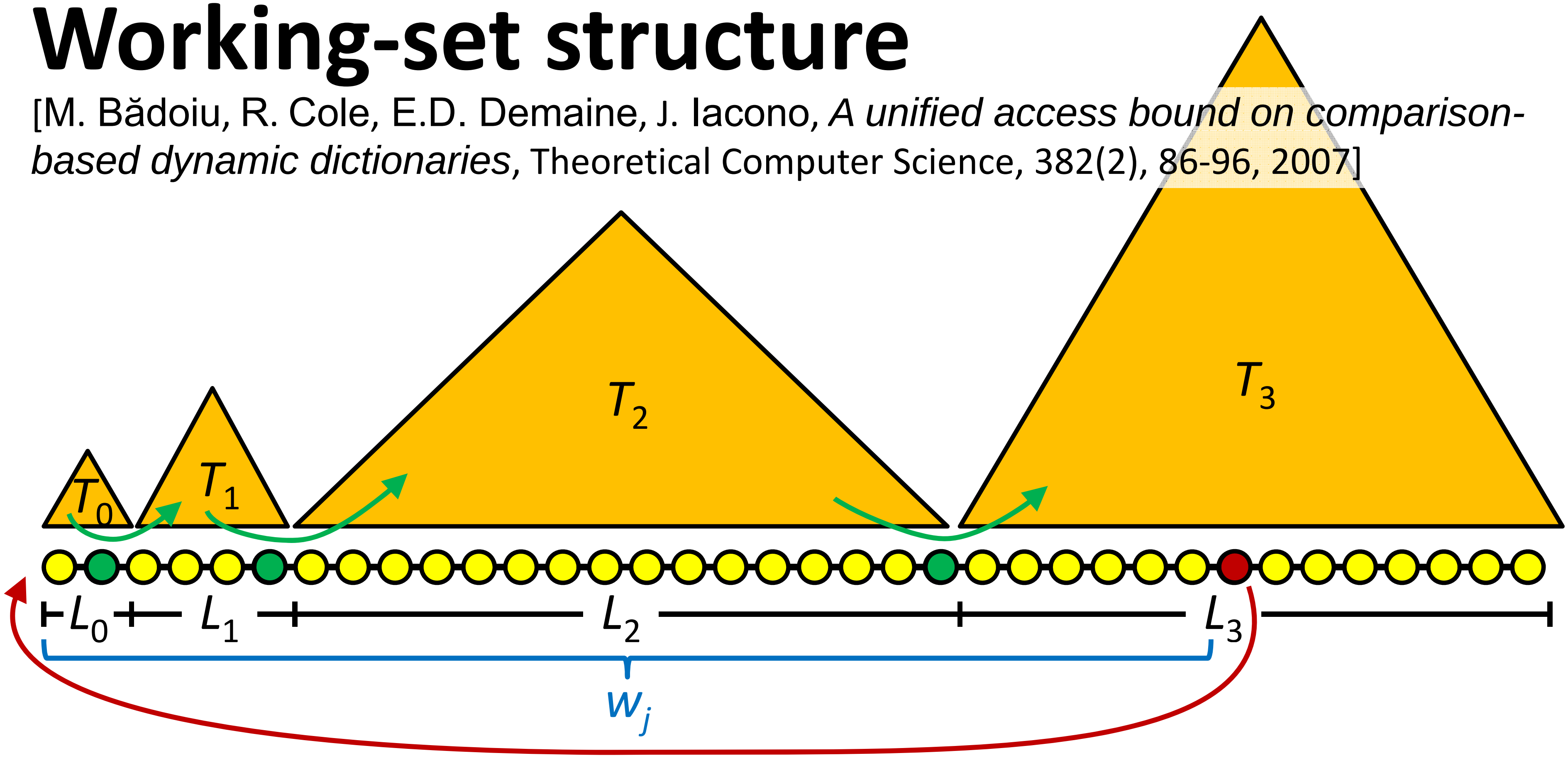
Split $\leq$ ½ weight both children
$\Rightarrow$ Depth i subtree weight $\leq (½)^i$
$\Rightarrow$ Depth $x_i \leq \log(1/p(x_i))$
$\Rightarrow$ Static optimal

x_i	A	B	C	D	E	F	G	H	I	J
$p(x_i)$	0.10	0.05	0.02	0.60	0.03	0.08	0.02	0.06	0.01	0.03

Construction: Compute prefix sums + Exponential search $\Rightarrow O(n)$

Working-set structure

[M. Bădoiu, R. Cole, E.D. Demaine, J. Iacono, *A unified access bound on comparison-based dynamic dictionaries*, Theoretical Computer Science, 382(2), 86-96, 2007]

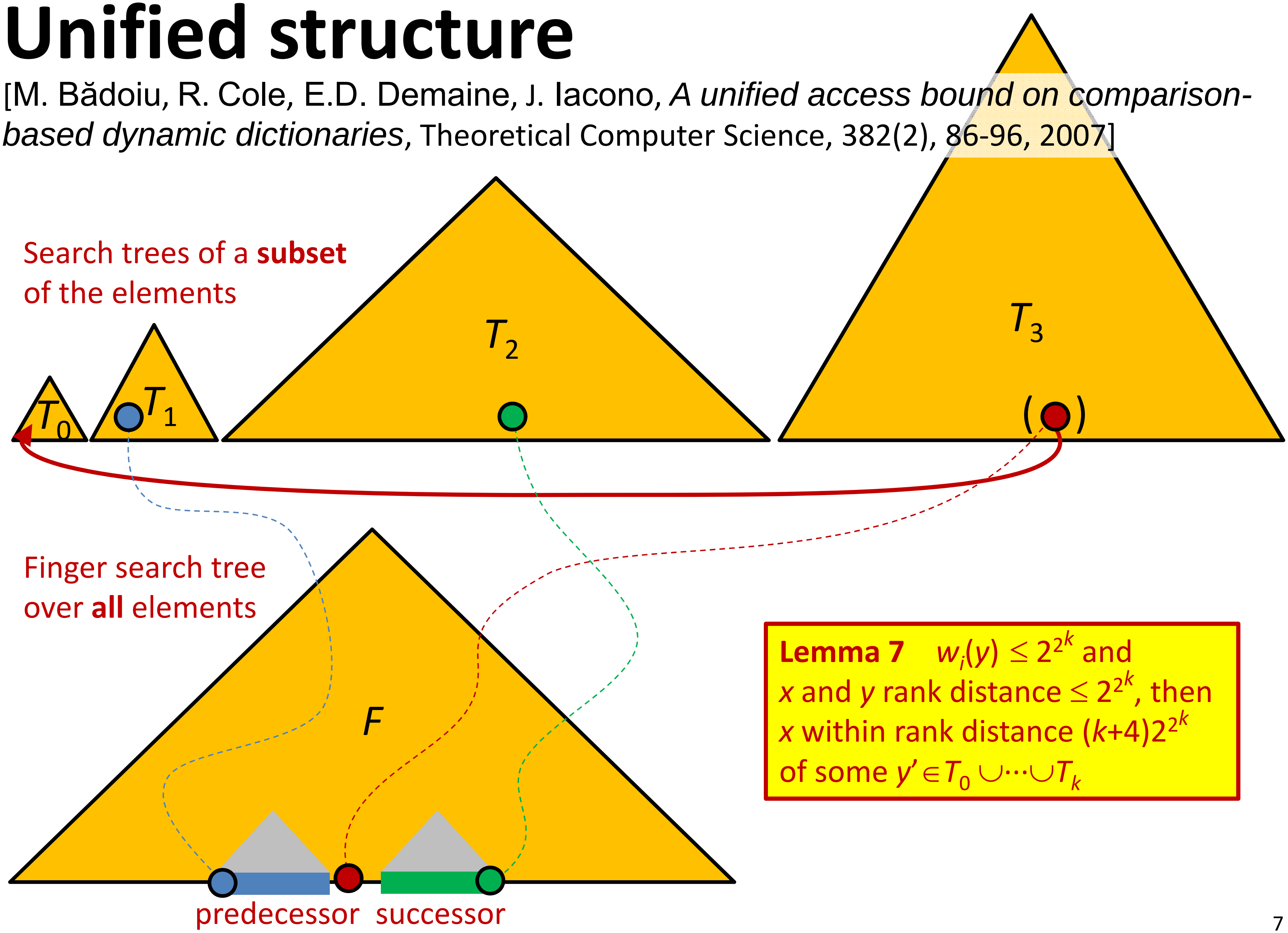


- $L = L_0 + L_1 + \cdots$ = move-to-front list
- $|L_i| = 2^{2^i}$
- T_i = search tree over L_i
- Insert, Delete = O(log n), Search = O(log w_j)

$$\boxed{\begin{array}{c}\text{Search}(T_0) + \cdots + \text{Search}(T_j) \\ = \log(2^{2^0}) + \cdots + \log(2^{2^j}) \\ \le 4\cdot\log(2^{2^{j-1}}) \le 4\cdot\log w_j\end{array}}$$

Unified structure

[M. Bădoiu, R. Cole, E.D. Demaine, J. Iacono, *A unified access bound on comparison-based dynamic dictionaries*, Theoretical Computer Science, 382(2), 86-96, 2007]



Lemma 7 $w_i(y) \leq 2^{2^k}$ and x and y rank distance $\leq 2^{2^k}$, then x within rank distance $(k+4)2^{2^k}$ of some $y' \in T_0 \cup \cdots \cup T_k$