Unified Access Bound


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

**Solution 1**: Balanced search tree $\Rightarrow$ $O(\log n)$

**Solution 2**: Unordered linked list $\Rightarrow$ $O(n)$


The paper initiated the study of competitiveness 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: $\mathcal{O}(\log (1/p(x_i)))$
- Sequential-access bound: $\mathcal{O}(1)$
- Static finger bound: $\mathcal{O}(\log d_i(f, x_i))$
- Dynamic finger bound: $\mathcal{O}(\log d_i(x_i, x_{i-1}))$
- Working set bound: $\mathcal{O}(\log w_i(x_i))$
- Unified bound: $\mathcal{O}(\min_{y \in S_i} \log(w_i(y) + d_i(x_i, y)))$

Sorted list

Move-to-front list
Splay trees (amortized)

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

Static optimality


### Construction:

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

### Split decision:

- Split \(\leq \frac{1}{2}\) weight both children
  - Depth \(d\) subtree weight \(\leq (\frac{1}{2})^d\)
  - Depth \(x_i \leq \log(1/p(x_i))\)
  - Static optimal
Working-set structure


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


Search trees of a **subset** of the elements

Finger search tree over **all** elements

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