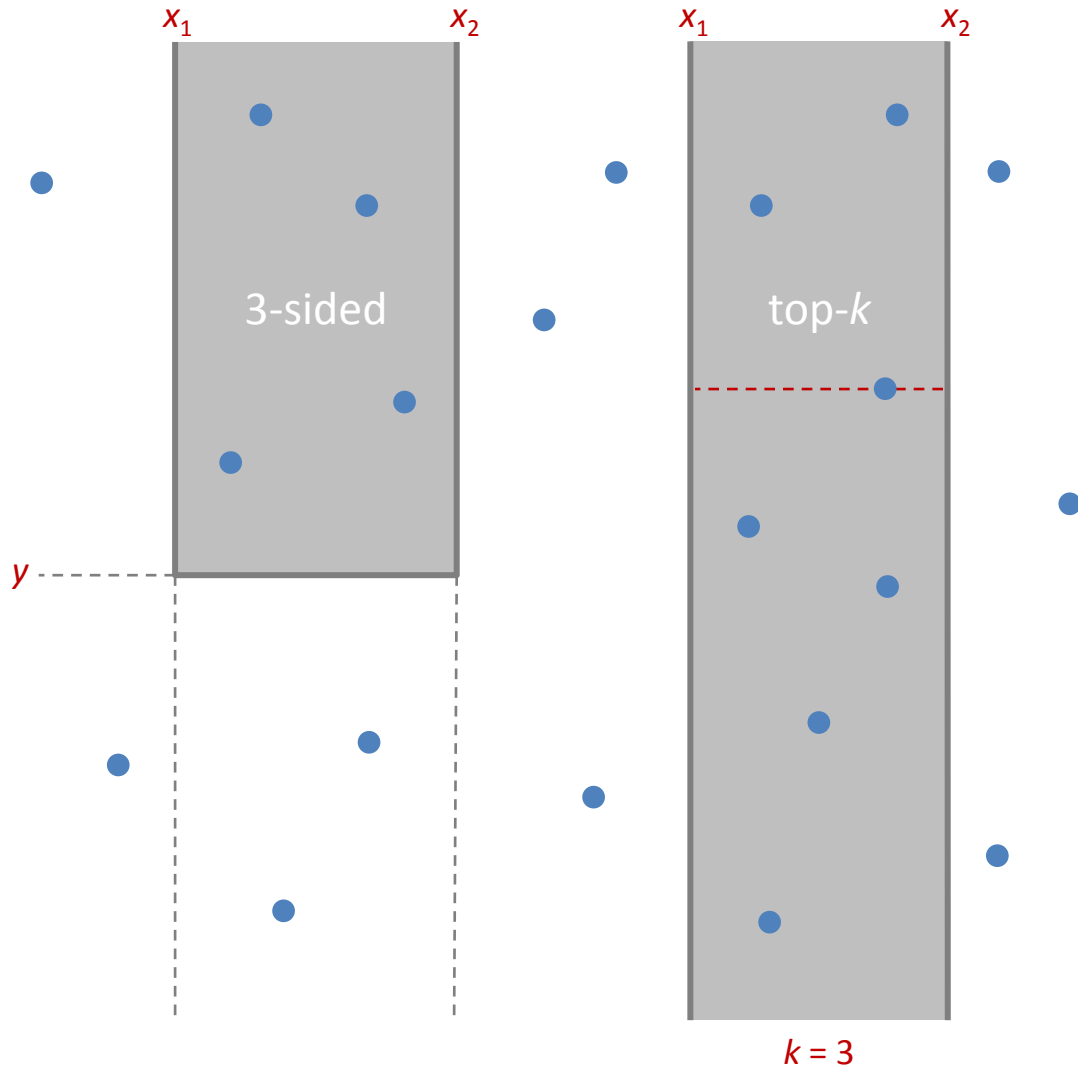


External Memory Three-Sided Range Reporting and Top- k Queries with Sublogarithmic Updates

arxiv.org/abs/1509.08240

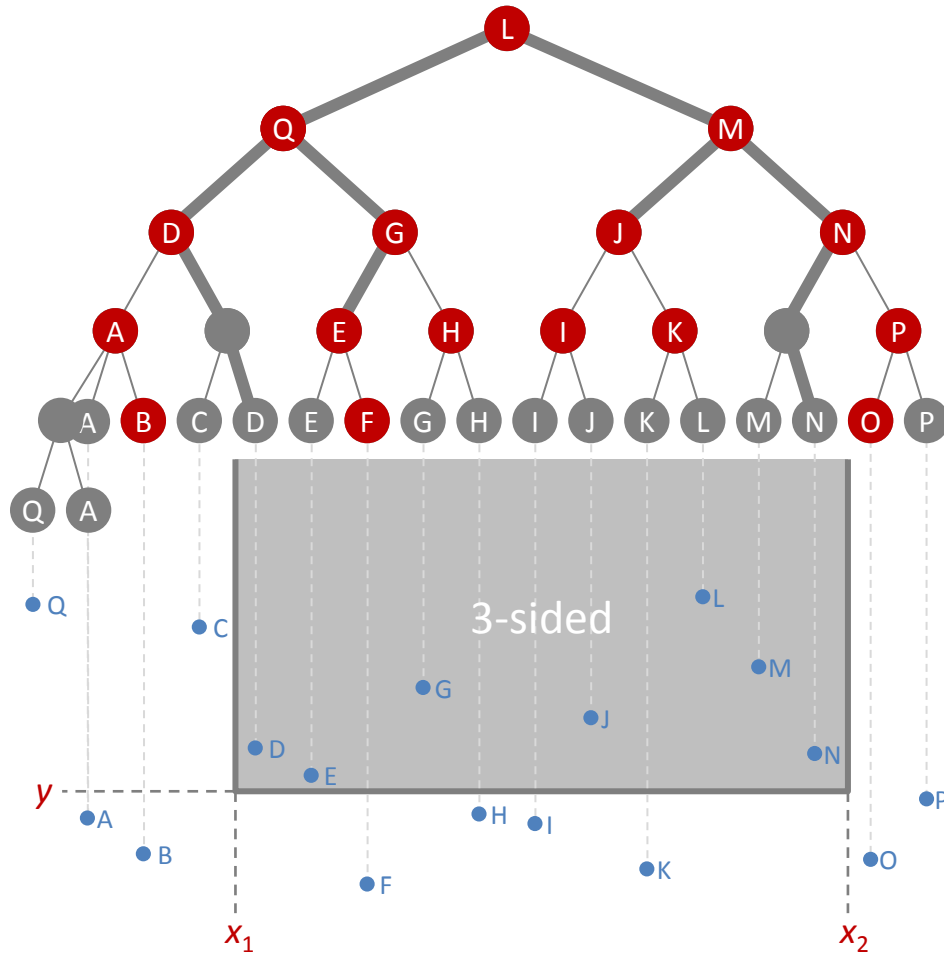


"the result is obtained by combining already existing techniques (and no new techniques are introduced)"

- anonymous reviewer

Internal Memory – Priority Search Trees

McCreight 1985
Frederickson 1993



Properties

- leaves x-sorted
- point p stored on leaf p -to-root path
- y -values satisfy heap-order

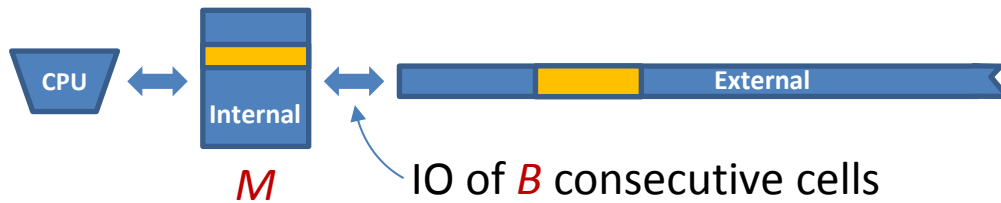
Updates $O(\log n)$
3-sided & top- k $O(\log n + k)$



select $k+s$ largest in $O(k+s)$ time

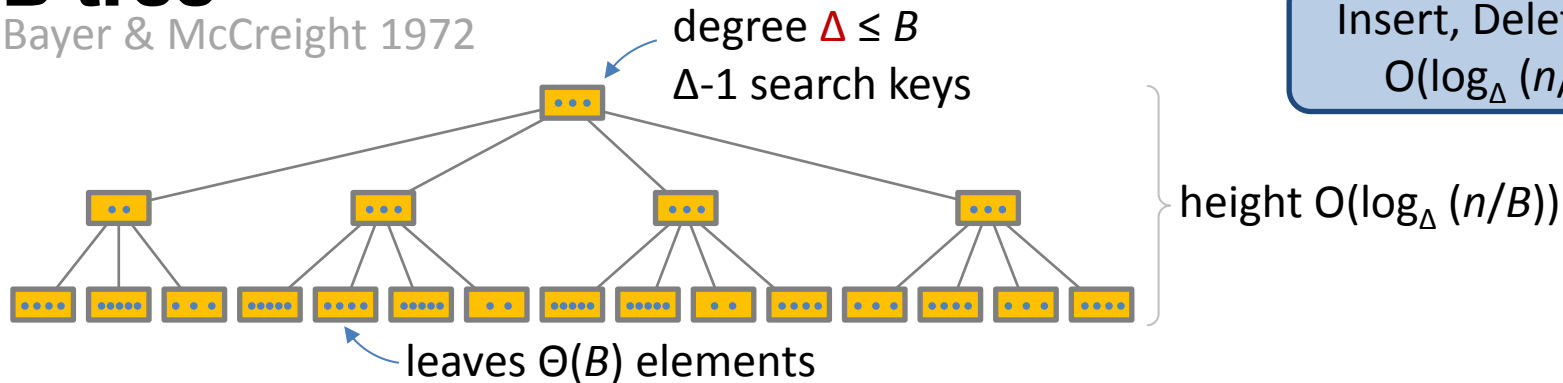
External Memory Model

Aggarwal & Vitter 1988



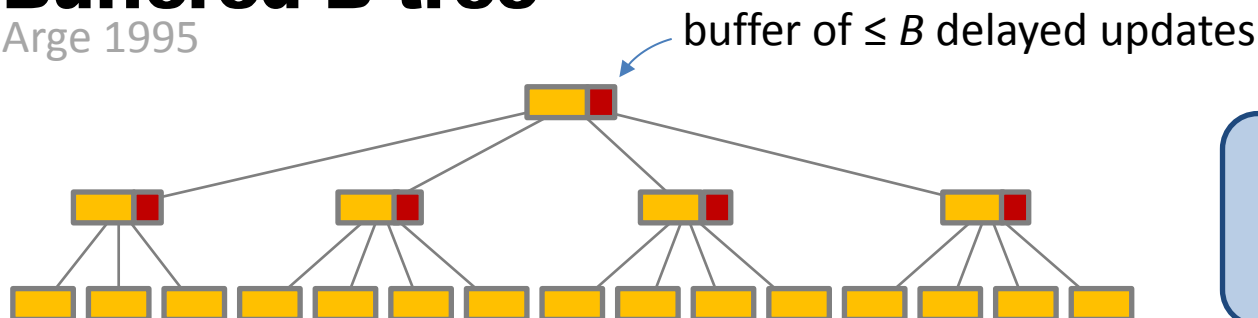
B-tree

Bayer & McCreight 1972



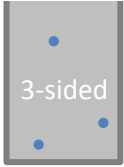
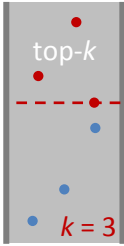
Buffered B-tree

Arge 1995



Search $O(\log_{\Delta} (n/B))$ IOs
Updates amortized
 $O(\Delta/B \cdot \log_{\Delta} (n/B))$ IOs

External Memory Results

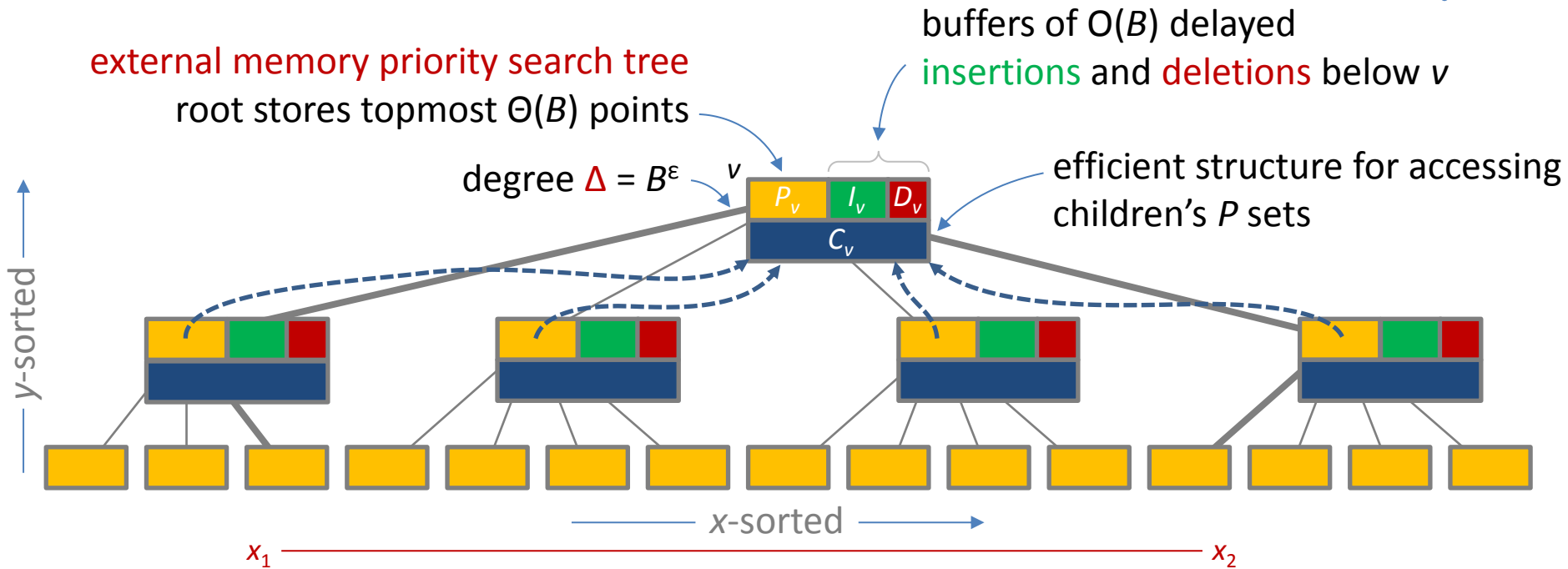
		Updates	Query
	Ramaswamy , Subramanian 1995	$O_A(\log n \cdot \log B)$	$O(\log_B n + k/B)$
	Subramanian, Ramaswamy 1995	$O_A(\log_B n + (\log_B n)^2/B)$	$O(\log_B n + k/B + \log^{**} B)$
	Arge et al. 1999	$O(\log_B n)$	$O(\log_B n + k/B)$
	STACS16	$O_A(1/(\epsilon B^{1-\epsilon}) \cdot \log_B n)$	$O_A(1/\epsilon \cdot \log_B n + k/B)$
	Afshani et al. 2011	(static)	$O(\log_B n + k/B)$
	Sheng, Tao 2012	$O_A((\log_B n)^2)$	$O(\log_B n + k/B)$
	Tao 2014	$O_A(\log_B n)$	$O(\log_B n + k/B)$
	STACS16	$O_A(1/(\epsilon B^{1-\epsilon}) \cdot \log_B n)$	$O_A(1/\epsilon \cdot \log_B n + k/B)$

O_A = amortized

NEW result : Combination of Arge 1995, Arge et al. 1999, Frederickson 1993, Blum et al. 1973

External Memory 3-sided Data Structure

3-sided

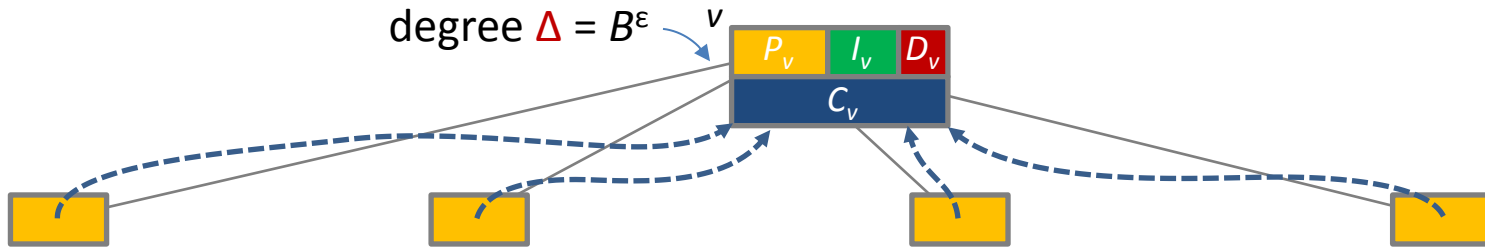


- **Insertions / deletions** : Update root P_v or add to delayed update buffer I_v / D_v
- Update buffer **overflow** : Flush recursively to child with most updates ($\geq B^{1-\epsilon}$)
- Leaf **overflow** : split leaf, and recursively split ancestors of degree $\Delta+1$
- **Underflowing** point buffer P_v : pull elements recursively from children using C_v
- **3-sided query** : *i)* Identify nodes to **visit** using C_v structures. *ii)* flush updates down from ancestors of visited nodes. *iii)* report from nodes using P_v , C_v and update buffers

Child Structure C_v

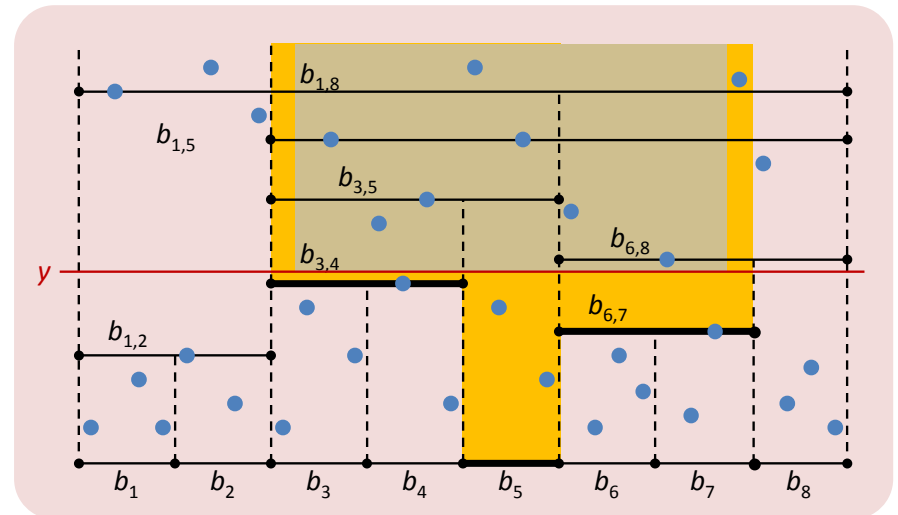
Arge et al. 1999

3-sided

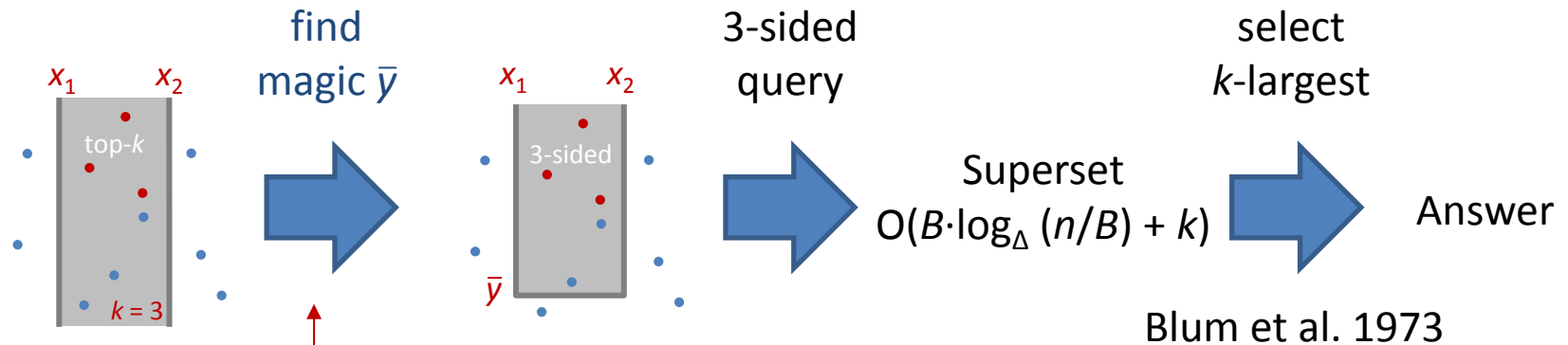


Insert / delete s points : $O(1 + s/B^{1-\epsilon})$ IOs
 3-sided query : $O(1 + k/B)$ IOs
 y -samples for range $[x_1, x_2]$: $O(1)$ IOs (new)

- Capacity : $B^{1+\epsilon}$
- Insetion /deletion buffer $O(B)$ points
- $O(B^\epsilon)$ blocks
- Catalog block
- y -samples block (new)



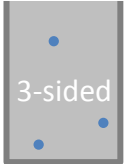
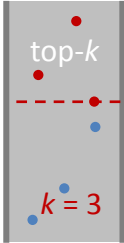
External Memory Top- k – Overall Approach



All steps require $O(\log_{\Delta}(n/B) + k/B)$ IOs
The 3-sided query is amortized

Construct (on demand) a **binary heap** over the samples of every $\Theta(B)$ 'th element in the C_v structures – and select the $\Theta(\log_{\Delta}(n/B) + k/B)$ 'th element using Frederickson 1993

Summary – The End

		Updates	Query
 <p>3-sided</p>	Ramaswamy , Subramanian 1995	$O_A(\log n \cdot \log B)$	$O(\log_B n + k/B)$
	Subramanian, Ramaswamy 1995	$O_A(\log_B n + (\log_B n)^2/B)$	$O(\log_B n + k/B + \log^{**} B)$
	Arge et al. 1999	$O(\log_B n)$	$O(\log_B n + k/B)$
	STACS16	$O_A(1/(\epsilon B^{1-\epsilon}) \cdot \log_B n)$	$O_A(1/\epsilon \cdot \log_B n + k/B)$
 <p>top-k</p> <p>k = 3</p>	Afshani et al. 2011	(static)	$O(\log_B n + k/B)$
	Sheng, Tao 2012	$O_A((\log_B n)^2)$	$O(\log_B n + k/B)$
	Tao 2014	$O_A(\log_B n)$	$O(\log_B n + k/B)$
	STACS16	$O_A(1/(\epsilon B^{1-\epsilon}) \cdot \log_B n)$	$O_A(1/\epsilon \cdot \log_B n + k/B)$

O_A = amortized

Open problem :
Remove amortization ?