On ‘Selection and Sorting with Limited Storage’

Graham Cormode

Joint work with S. Muthukrishnan, Andrew McGregor, Amit Chakrabarti

Mike Paterson

SELECTION AND SORTING WITH LIMITED STORAGE

J.I. Munro *

Dept. of Computer Science
University of Waterloo
Ontario, Canada.

M.S. Paterson

Dept. of Computer Science
University of Warwick
Coventry, U.K.

Abstract

When selecting from, or sorting, a file stored on a read-only tape and the internal storage is rather limited, several passes of the input tape may be required. We study the relation between the amount of internal storage available and the number of passes required to select the K^{th} highest of N inputs. We show, for example, that to find the median in two passes requires at least $\Omega(N^{\frac{1}{2}})$ and at most $O(N^{\frac{1}{2}} \log N)$ internal storage. For probabilistic methods, $\theta(N^{\frac{1}{2}})$ internal storage is necessary and sufficient for a single pass method which finds the median with arbitrarily high probability.

storage. The elements are from some totally ordered set (for example the real numbers) and a binary comparison can be made at any time between any two elements within the random-access storage. Initially the storage is empty and the tape is placed with the reading head at the beginning. After each pass the tape is rewound to this position with no reading permitted.

Notational note.

For functions of several arguments we shall write $f(\chi) = O(g(\chi))$ when $\exists c > 0$ such that $|f(\chi)| < c.g(\chi)$ for all χ except those naturally or explicitly excluded. We also use $f = \theta(g)$ for $g = O(f)$; and we use $f = \Omega(g)$ for $f = O(g)$ and $g = O(f)$.

Munro-Paterson 78 [MP78]

- One of the first papers to consider computing with limited storage
  - Storage sublinear in the size of the input
- Considered what could be accomplished in one or few passes over input treated as a one-way tape
  - Effectively the now-popular ‘streaming model’
- Focused on the problem of selection (median and generalized median)
  - ‘Selection’: Find the K’th ranked item (integer) out of N
  - Dozens of papers on variations of these problems in the streaming world in last decade

Results in MP78

- P-pass deterministic algorithm for selection
  - In each pass, narrow down the range of interest
  - Compute the exact ranks of a small range in the final pass
  - Recursively merge and thin out pairs of buffers, tracking bounds on the ranks of each retained item
  - Gives $O(N^{1/P}$ poly-log$(N))$ space for P passes
    - Implies $P=\log N$ passes in poly-log(N) space
- Revisited by Manku, Rajagopalan and Lindsay [1998]:
  - Obtain εN error in ranks in $O(\varepsilon^{-1}\log^2 \varepsilon N)$ space, one pass
  - Improved to $O(\varepsilon^{-1}\log \varepsilon N)$ by Greenwald and Khanna

Results in MP78

- Deterministic lower bound of $\Omega(N^{1/P})$ space for P-passes
  - Based on an adversary who ensures that there are many elements not stored whose relative ordering is unknown
- Later, $\Omega(1/\varepsilon)$ bound for one pass approximate selection allowing randomness—implies $\Omega(N)$ bound for exact
  - Shown by Henzinger, Raghavan, Rajagopalan [1998]

Results in MP78

- Bounds "assuming that all input orderings are equally likely"
  - Now known as the "random order streams assumption"
    - Shows a problem is hard even under favourable order
  - $O(N^{1/2P})$ upper bound, and $\Omega(N^{1/2})$ 1 pass lower bound
- Guha and McGregor [2006] give a $P=O(\log \log N)$ pass algorithm for exact selection in $O(\text{polylog}(N))$ space
  - An exponential gap between the adversarial order case
  - Resolves a question posed in MP78.
  - Is this optimal?

Outline

- Selection and Sorting with Limited Storage
- One pass approximate selection with deletions
- Lower bounds for P pass selection on random order input

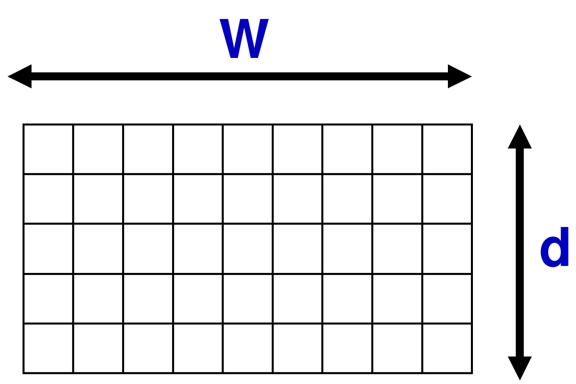
Approximate Selection with Deletions

- ε-approximate selection:
  – Find any item with rank between (Φ-ε)N and (Φ+ε)N
- Streams with deletions:
  – Stream contains both "insertion" and "deletion" of items
  – Assume no deletions without preceding matching insertion
  – Captures e.g. database transactions, network connections
- Assumption: items drawn from bounded universe of size U
  – Model as integers 1…U
- Approach: solve a different streaming problem, then reduce
  – Estimate frequency of some item j with additive error εN

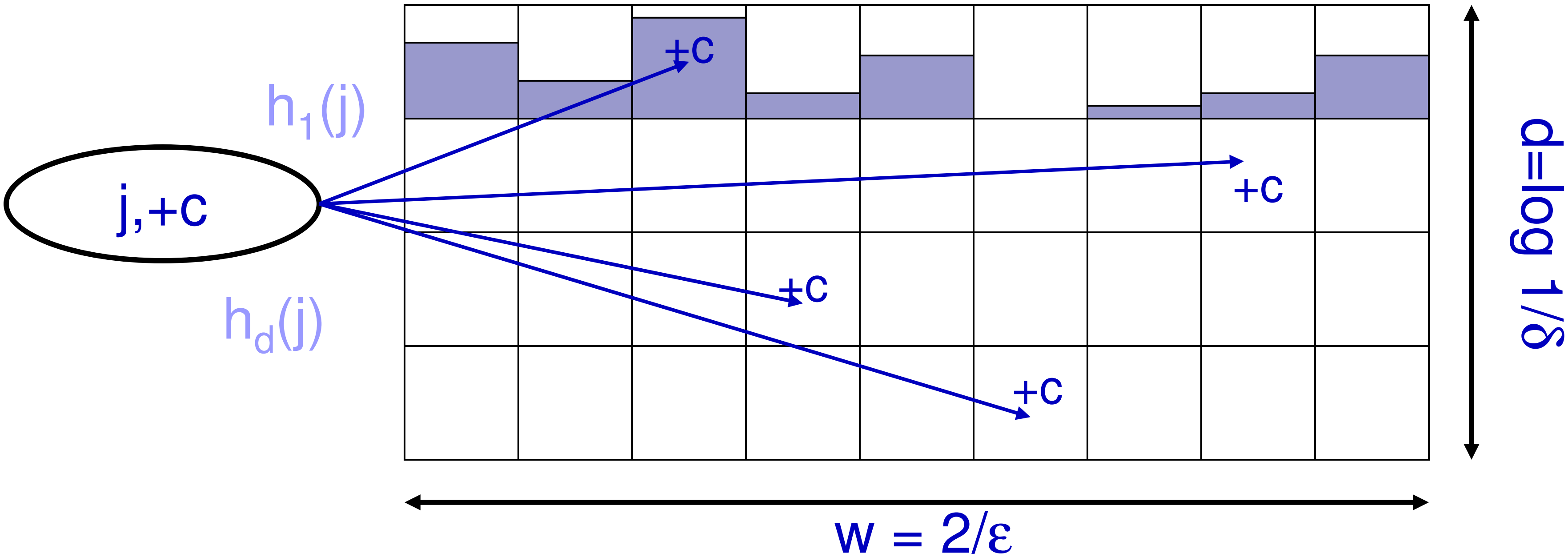
Count-Min Sketch

- Simple sketch idea, can be used for as the basis of many different stream analysis.
- Model input stream as a vector x of dimension U
- Creates a small summary as an array of $w \times d$ in size
- Use d hash function to map vector entries to [1..w]
- Works on arrivals only and arrivals & departures streams

Array:
CM[i,j]



CM Sketch Structure



- Each entry in vector x is mapped to one bucket per row.
- Merge two sketches by entry-wise summation
- Estimate x[j] by taking $\min_k CM[k,h_k(j)]$
  - Guarantees error less than $\varepsilon||x||_1$ in size $O(1/\varepsilon \log 1/\delta)$
  - Probability of more error is less than $1-\delta$

[C, Muthukrishnan '04]

Approximation

Approximate $x'[j] = \min_k CM[k,h_k(j)]$

- Analysis: In k'th row, $CM[k,h_k(j)] = x[j] + X_{k,j}$
  - $X_{k,j} = \Sigma\, x[i] \mid h_k(i) = h_k(j)$
  - $E(X_{k,j}) = \Sigma\, x[k]^*Pr[h_k(i)=h_k(j)]$
    $\leq Pr[h_k(i)=h_k(k)] * \Sigma\, a[i]$
    $= \varepsilon\, ||x||_1/2$ by pairwise independence of h
  - $Pr[X_{k,j} \geq \varepsilon ||x||_1] = Pr[X_{k,j} \geq 2E(X_{k,j})] \leq 1/2$ by Markov inequality
- So, $Pr[x'[j] \geq x[j] + \varepsilon\, ||x||_1] = Pr[\forall\ k.\ X_{k,j} > \varepsilon\, ||x||_1] \leq 1/2^{\log 1/\delta} = \delta$
- Final result: with certainty $x[j] \leq x'[j]$ and
  with probability at least $1-\delta$, $x'[j] < x[j] + \varepsilon\, ||x||_1$

Application To Selection

- Impose a binary tree over the domain of input items
  - Each node corresponds to the union of its leaves
- Keep a CM sketch to summarize each level of the tree
- Estimate the rank of any item from $O(\log U)$ dyadic ranges and estimate each from relevant sketch

- For selection, binary search over the domain of items to find one with the desired estimated rank
- Result: solve one-pass ε-approximate selection with probability at least $1-\delta$ using $O(1/\varepsilon \log^2 U \log 1/\delta)$ space
  - Deterministic solution requires $\Omega(1/\varepsilon^2)$ space

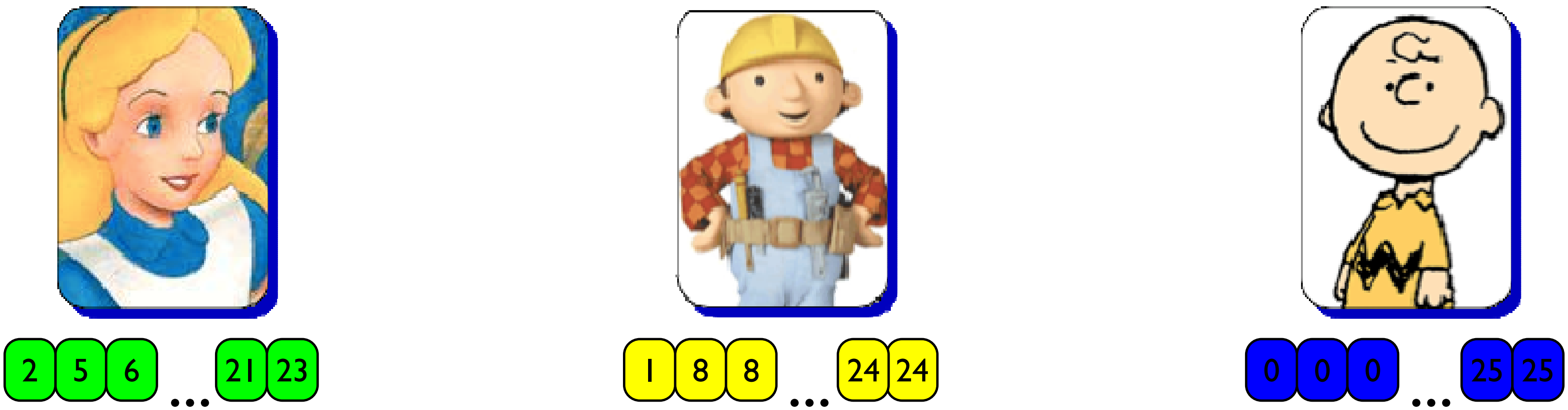
Outline

- Selection and Sorting with Limited Storage
- One pass approximate selection with deletions
- Lower bounds for P pass selection on random order input

Bounds Via Communication Complexity

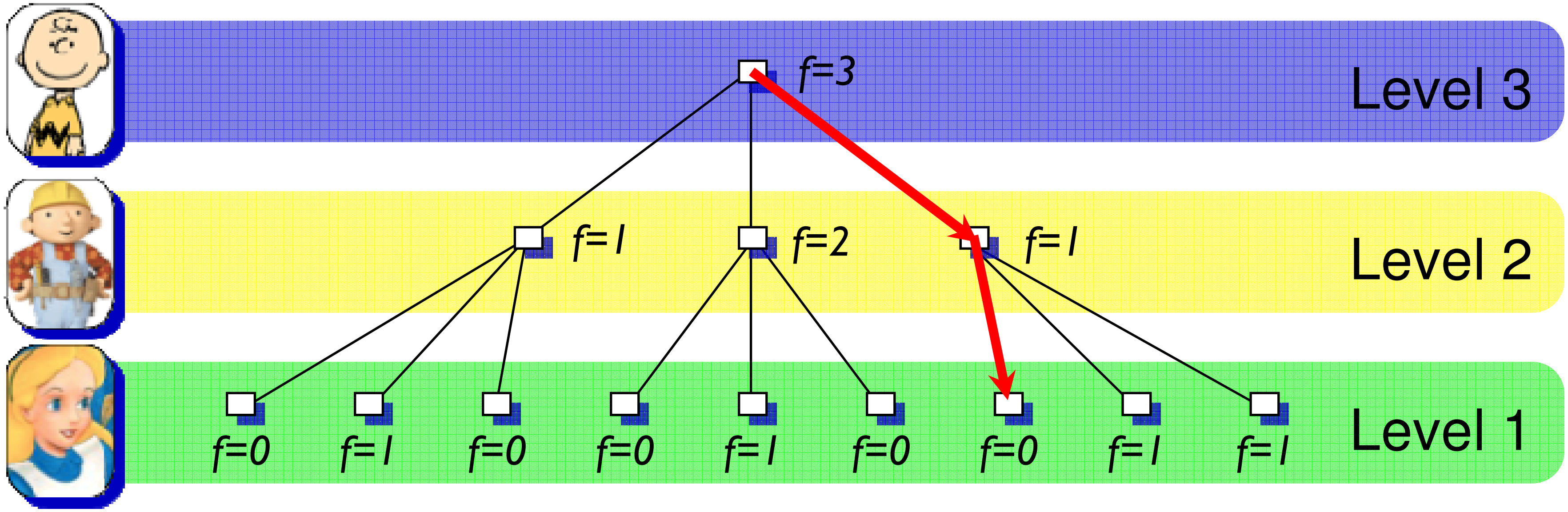
- Viewing contents of memory as a *message* being passed, communication complexity techniques give space lower bounds
  - Sending the contents of memory gives a communication protocol
  - Similar style of argument used in [MP78] to bound space of a P-pass sorting algorithm
- Proving lower bounds for streams in random order led us to consider communication bounds for random partitions of the input between players [Chakrabarti, C, McGregor 08]

The Model



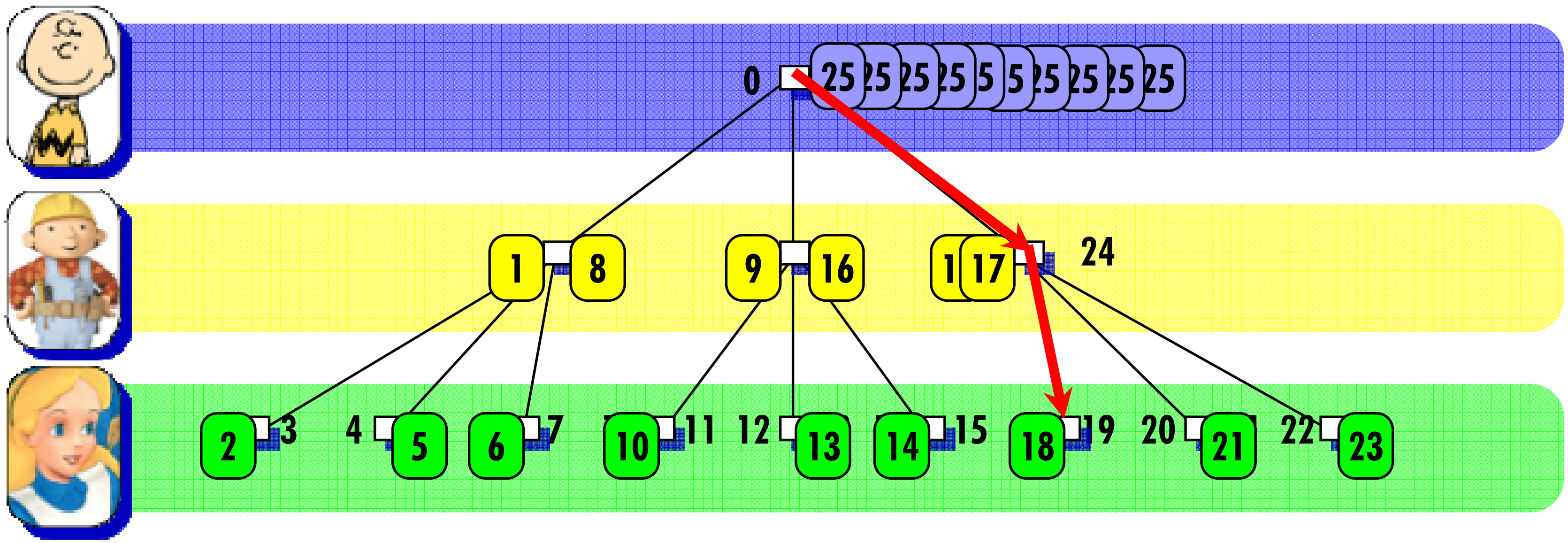
- The P players (Alice, Bob, Charlie…) each receive a random partition of input (could be non-uniform)
- Each communicates a message in order to the next, in up to r rounds
- Lower bounds on communication imply streaming space lower bounds

Tree Pointer Jumping (TPJ)



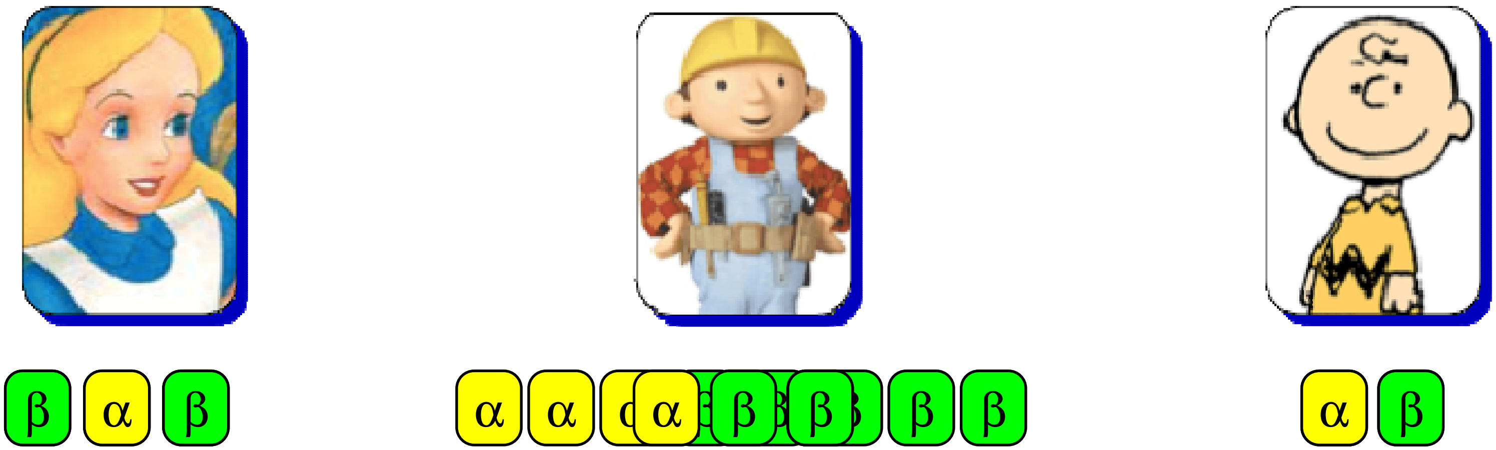
- Instance: Function on nodes of P-level, t-ary tree,
  - if v is an internal node: f maps v to a child of v
  - if v is a leaf: f maps v to {0,1}
- Goal: Compute $f(f(\ldots f(v_{root})\ldots))$.
- For P-players, if i^{th} player knows f(v) when level(v)=i:
  Any (P-1)-round protocol requires $\Omega(t/P^2)$ communication.
  - Even when input is picked uniformly at random

Reduction from TPJ to Median



- With each node v associate two values $\alpha(v) < \beta(v)$ such that $\alpha(v) < \alpha(u) < \beta(u) < \beta(v)$ for any descendent u of v.
- For each node: Generate more copies of $\alpha(v)$ and $\beta(v)$ such that median of values corresponds to TPJ solution.
- Relationship between t and # copies determines bound.
  - Need more copies higher up in tree

Simulating Random-Partition Protocol



- Consider tree node v where f(v) is known to Bob.
- Create Instance of Random-Partition Tree-Pointer Jumping:
  1) Using public coin, players determine partition of tokens and set half to α and half to β.
  2) Bob "fixes" balance of tokens under his control.
- The resulting distribution is "close" so algorithm expecting a random partition should succeed with only slightly lower prob

Implications for Selection

- Implies a communication lower bound of $\Omega(N^{1/2^P})$ (supressing lesser factors)
- Means any P-pass algorithm for median finding (more generally, selection) requires $\Omega(N^{1/2^P}/2^P)$ space
  - poly(log N) space requires $P=\theta(\log \log N)$ passes
  - 3 pass algorithm requires $\Omega(N^{1/10})$ space

Conclusions

- ‘Selection and Sorting with Limited Storage’ continues to be an influential paper, three decades later.
- Several related papers accepted to SODA 2009:
  - Comparison-Based, Time-Space Lower Bounds for Selection (Timothy M. Chan)
  - Sorting and Selection in Posets (Constantinos Daskalakis, Richard M. Karp, Elchanan Mossel, Samantha Riesenfeld and Elad Verbin)