Fragile Complexity of Comparison-Based Algorithms

Würzburg, Informatik-Kolloquium, Sept. 26, 2019
Best Paper Track A ESA 2019

Riko Jacob
IT University of Copenhagen, DK

Joint work with
Peyamn Afshani, Rolf Fagerberg, David Hammer,
Irina Kostitsyna, Ulrich Meyer, Manuel Penschuck,
Nodari Sitchinava

Destructive Comparisons

- sports competition

Destructive Comparisons

- sports competition
- comparing beers

Fragile Complexity

Algorithm

- Comparison Based (sorting, minimum, median/selection)

Fragile Complexity

$f(n)$ maximal number of comparisons any single element participates in

$f_m(n)$ number of comparisons the minimum/selected element participates in

$f_{rem}(n)$ like $f(n)$, bound for any element

$w(n)$ work, total number of comparisons

Observe: All of them between 1 and $n - 1$

Questions

Sorting

Questions

Sorting

- Deterministic: AKS sorting networks

Questions

Sorting

- Deterministic: AKS sorting networks
- Randomized merge sort

Questions

Sorting

- Deterministic: AKS sorting networks
- Randomized merge sort

Minimum

Protect the winner!

Considered Variants

Deterministic

Randomized

- The reported result is always correct (Las Vegas)
- expected value $E[f_m(n)]$
- $E[f_{rem}(n)]$ (expectation of max, **NOT** max of expectation)

Overview Results

Problem		*Upper*		*Lower*
		$f(n)$	$w(n)$	$f(n)$
MINIMUM	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\langle \mathcal{O}(1)^{\dagger}, n^{\varepsilon} \rangle$ $\langle \mathcal{O}(\log_{\Delta} n)^{\dagger}, \mathcal{O}(\Delta + \log_{\Delta} n)^{\dagger} \rangle$	$\mathcal{O}(n)$	$\langle \Omega(\log_{\Delta} n)^{\dagger}, \Delta \rangle$
		$\langle O(\log_{\Delta} n \log\log \Delta),$ $O(\Delta + \log_{\Delta} n \log\log \Delta) \rangle^{\ddagger}$	$\mathcal{O}(n)$	$\Omega(\log\log n)^{\ddagger}$
SELECTION	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\left\langle \mathcal{O}(\log\log n)^{\dagger}, \mathcal{O}\left(\sqrt{n}\right)^{\dagger} \right\rangle$ $\left\langle \mathcal{O}\left(\frac{\log n}{\log\log n}\right)^{\dagger}, \mathcal{O}(\log^2 n)^{\dagger} \right\rangle$	$\mathcal{O}(n)^{\dagger}$	$\langle \Omega(\log_{\Delta} n)^{\dagger}, \Delta \rangle$

$f(n)$ – fragile complexity; $w(n)$ – work; $\langle f_m(n), f_{rem}(n) \rangle$ – fragile complexity for the selected element (minimum/median) and the remaining elements, respectively; † – holds in expectation, ‡ – holds with high probability.

Deterministic Minimum

$\Theta(\log n)$ fragile complexity (for the min)

- Balanced Tournament Tree
- Adversary

Overview Results

Problem		*Upper* $f(n)$	*Upper* $w(n)$	*Lower* $f(n)$
MINIMUM	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\langle \mathcal{O}(1)^{\dagger}, n^{\varepsilon} \rangle$ $\langle \mathcal{O}(\log_{\Delta} n)^{\dagger}, \mathcal{O}(\Delta + \log_{\Delta} n)^{\dagger} \rangle$	$\mathcal{O}(n)$	$\langle \Omega(\log_{\Delta} n)^{\dagger}, \Delta \rangle$
		$\langle O(\log_{\Delta} n \log\log \Delta),$ $O(\Delta + \log_{\Delta} n \log\log \Delta) \rangle^{\ddagger}$	$\mathcal{O}(n)$	$\Omega(\log\log n)^{\ddagger}$
SELECTION	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\left\langle \mathcal{O}(\log\log n)^{\dagger}, \mathcal{O}\left(\sqrt{n}\right)^{\dagger} \right\rangle$ $\left\langle \mathcal{O}\left(\frac{\log n}{\log\log n}\right)^{\dagger}, \mathcal{O}(\log^2 n)^{\dagger} \right\rangle$	$\mathcal{O}(n)^{\dagger}$	$\langle \Omega(\log_{\Delta} n)^{\dagger}, \Delta \rangle$

$f(n)$ – fragile complexity; $w(n)$ – work; $\langle f_m(n), f_{rem}(n) \rangle$ – fragile complexity for the selected element (minimum/median) and the remaining elements, respectively; $\dagger$ – holds in expectation, $\ddagger$ – holds with high probability.

Overview Results

Problem		*Upper* $f(n)$	*Upper* $w(n)$	*Lower* $f(n)$
MINIMUM	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\langle \mathcal{O}(1)^\dagger, n^\varepsilon \rangle$ $\langle \mathcal{O}(\log_\Delta n)^\dagger, \mathcal{O}(\Delta + \log_\Delta n)^\dagger \rangle$	$\mathcal{O}(n)$	$\langle \Omega(\log_\Delta n)^\dagger, \Delta \rangle$
		$\langle O(\log_\Delta n \log\log \Delta),$ $O(\Delta + \log_\Delta n \log\log \Delta) \rangle^\ddagger$	$\mathcal{O}(n)$	$\Omega(\log\log n)^\ddagger$
SELECTION	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\left\langle \mathcal{O}(\log\log n)^\dagger, \mathcal{O}\left(\sqrt{n}\right)^\dagger \right\rangle$ $\left\langle \mathcal{O}\left(\frac{\log n}{\log\log n}\right)^\dagger, \mathcal{O}(\log^2 n)^\dagger \right\rangle$	$\mathcal{O}(n)^\dagger$	$\langle \Omega(\log_\Delta n)^\dagger, \Delta \rangle$

$f(n)$ – fragile complexity; $w(n)$ – work; $\langle f_m(n), f_{rem}(n) \rangle$ – fragile complexity for the selected element (minimum/median) and the remaining elements, respectively; † – holds in expectation, ‡ – holds with high probability.

Simple Sampling Algorithm

Algorithm to compute rank 1 and 2 elements

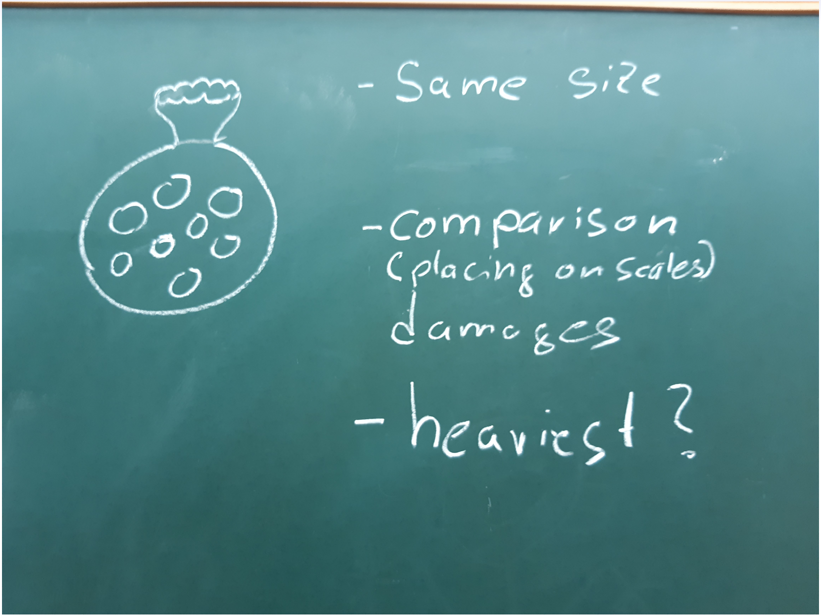
- Sample elements independently $p = 1/2$: Q (non-sampled O)
- Recursively find m_Q (rank 1) and s_Q (rank 2) elements in Q
- Filter O with s_Q: $O' = \{x \in O \mid x < s_Q\}$
- Find two smallest m_O, s_O elements of O' (directly)
- Comparison of m_Q and m_O, and $s.$ identifies output

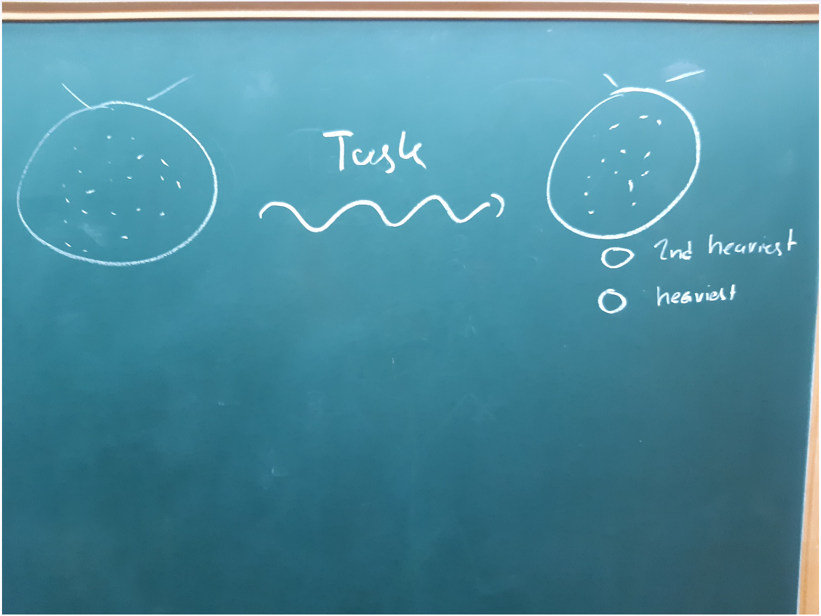
Analysis

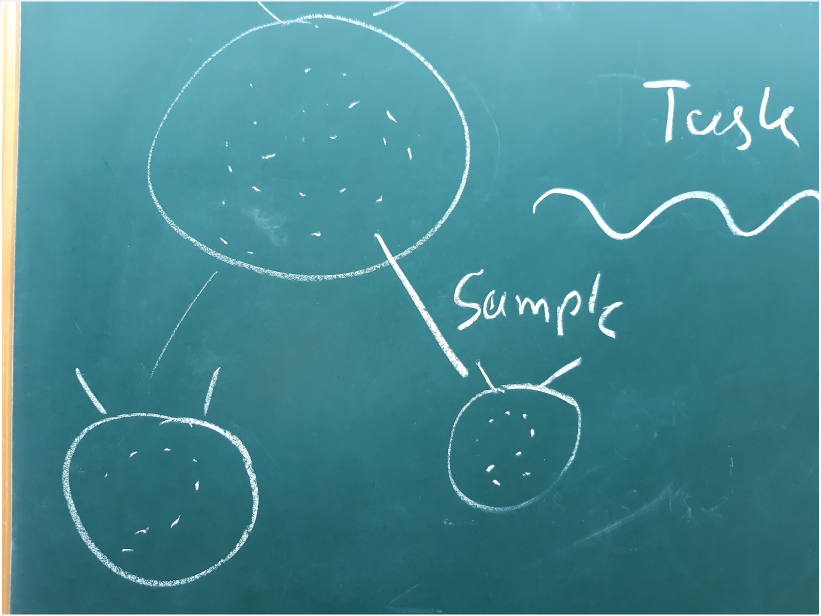
- $E[|O'|] = \sum_i 1/2^i = \mathcal{O}(1)$
- $f_m(n) = |O'| + \sum_i 1/2^i = \mathcal{O}(1)$

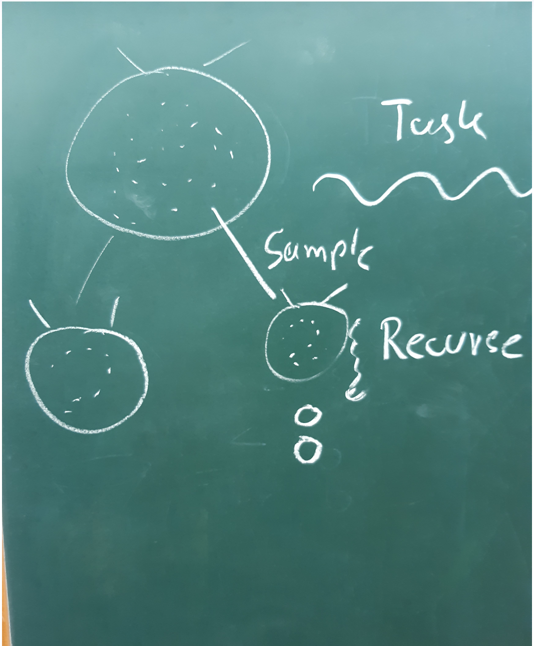
Tight Analysis

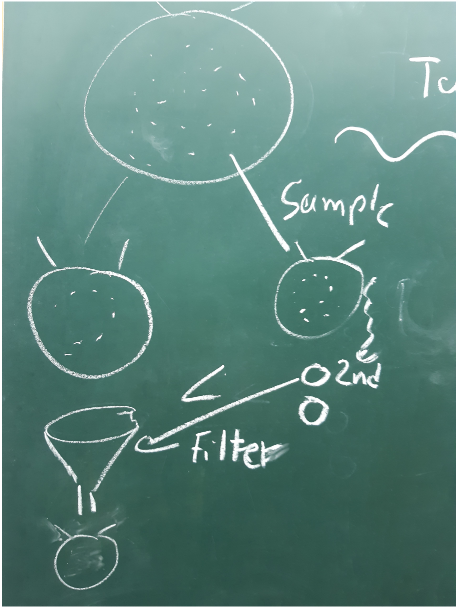
- $E[f(n)] = \Theta(n)$
- $\text{Prob}[f_m(n) > .5 \log n] = \Omega(n^{-.5})$

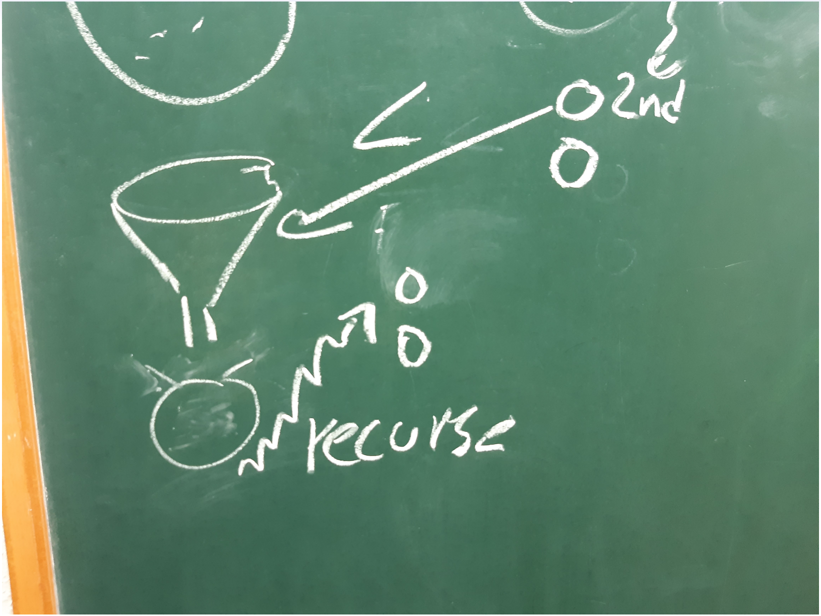

- Same size
- comparison
(placing on scales)
damages
- heaviest?

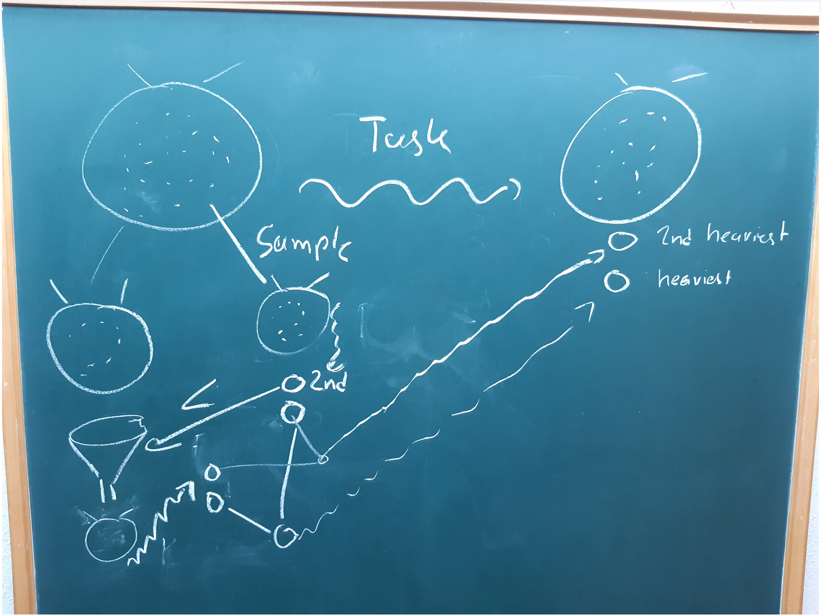

Task
2nd heaviest
heaviest

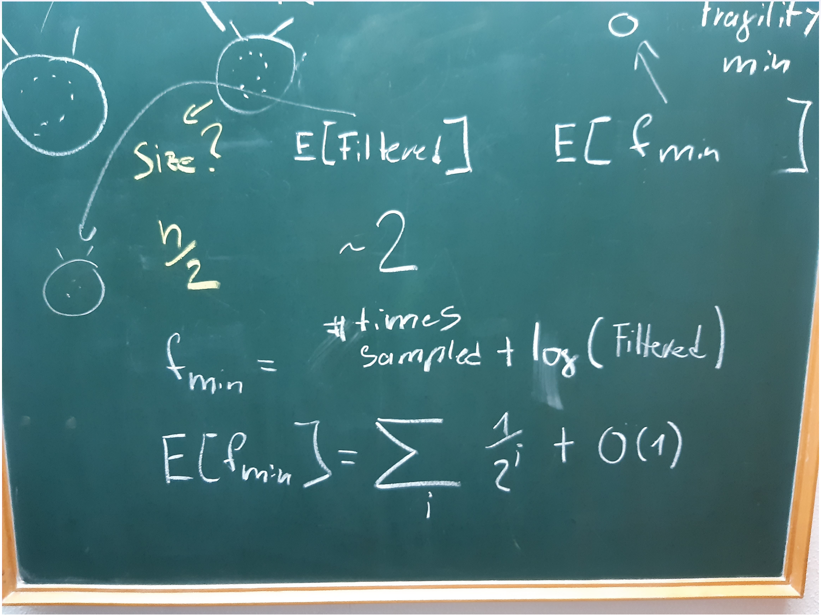

Task
Sample

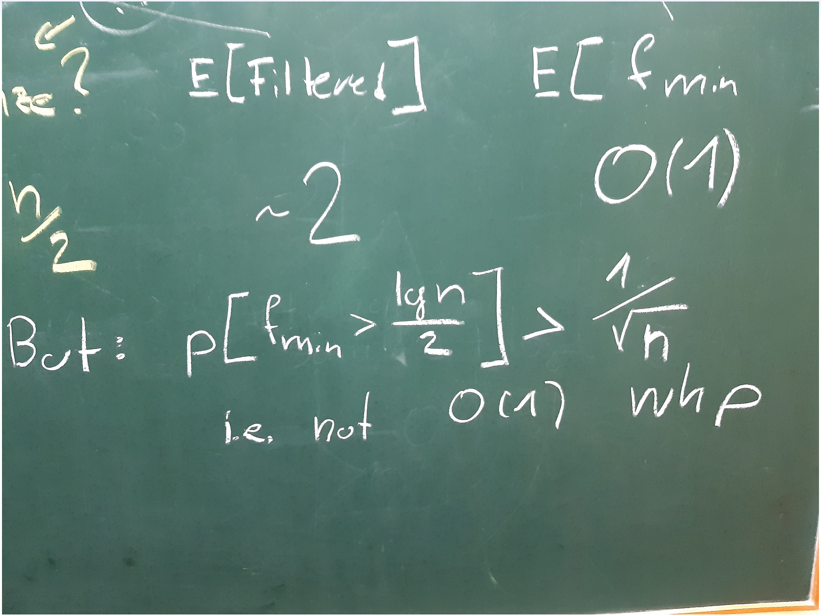

Task
Sample
Recurse

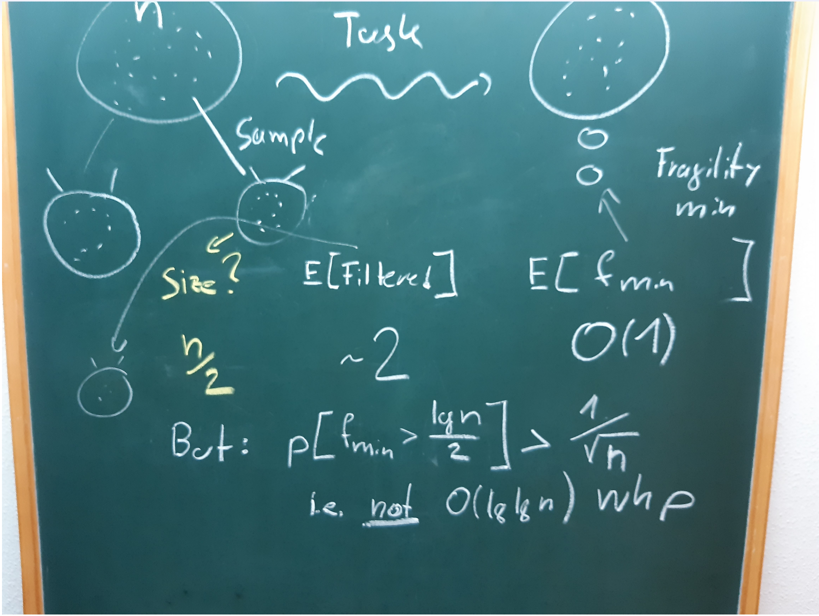

Sample
2nd
Filter

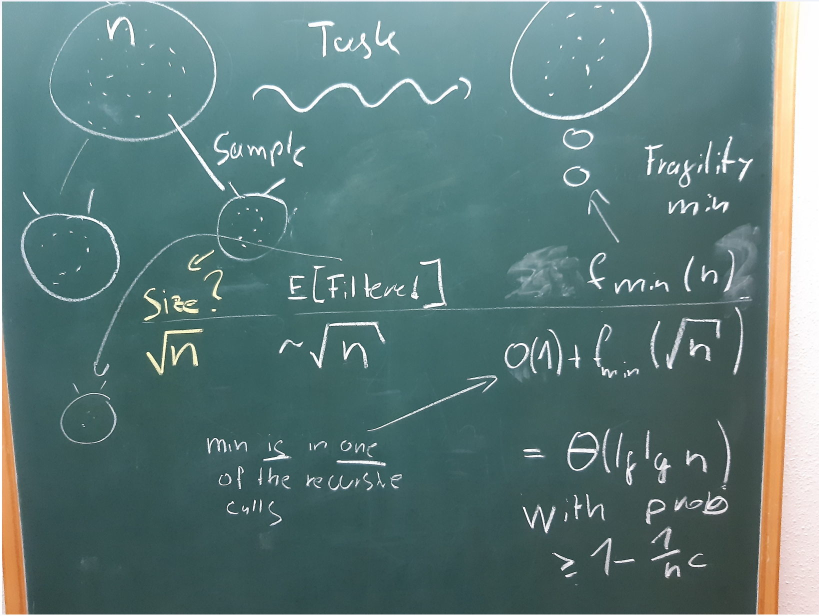

2nd
recurse

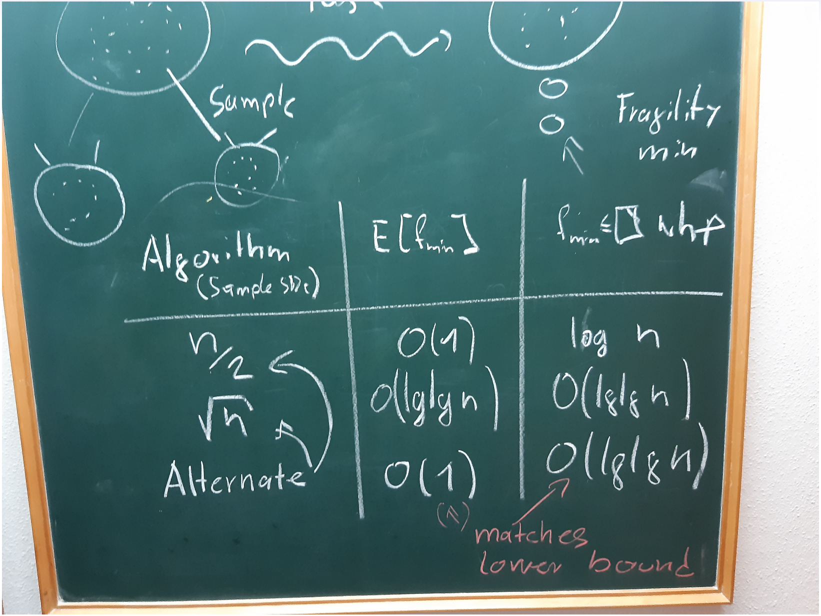

Task
Sample
2nd
2nd heaviest
heaviest


fragility
min
Size?
E[Filtered]
E[f_min]
n/2
~2
f_min = # times sampled + log (Filtered)
E[f_min] = ∑_i 1/2^i + O(1)


E[Filtered]
E[f_min
~2
O(1)
But: P[f_min > lg n / 2] > 1/√n
i.e. not O(1) whp


n
Task
Sample
Fragility min
Size?
E[Filtered]
E[f_min]
n/2
~2
O(1)
But: p[f_min > lg n / 2] > 1/√n
i.e. not O(lg lg n) whp


n
Task
Sample
Fragility
min
Size?
E[Filtered]
f min (n)
√n
~√n
O(1) + f min (√n)
min is in one
of the recursive
calls
= Θ(lg lg n)
with prob
≥ 1 − 1/n^c


Sample
Fragility
min
Algorithm
(Sample size)
E[f_min]
f_min ≤ [ ] whp
n/2
√n
Alternate
O(1)
O(lglg n)
O(1)
log n
O(lglg n)
O(lglg n)
matches
lower bound

Saving the Other Elements

Fragile complexity of the second smallest

The second smallest sampled element is in $\Omega(n)$ comparisons

Algorithm with $f(n) \leq \Delta$

- Randomly Partition into Δ sets S_i of size n/Δ
- solve S_i recursively, leading to $M = \{m_1, \ldots, m_\Delta\}$
- Find $\min(M)$ using the above algorithm

Analysis

- $E[f_m(n)] = O(\log_\Delta n)$ — $O(1)$ per recursion
- $f(n) \leq \Delta$ — “other element” only once

Examples for Δ

$f_{rem}(n) \leq \Delta = n^{-d}$

$E[f_m(n)] = \mathcal{O}(\log_\Delta(n)) = \mathcal{O}(1)$, still optimal

$\Delta = \log n / \log\log n$

$E[f_m(n)] = \mathcal{O}(\log_\Delta n) \leq_{a.e.} f_{rem}(n) \leq \Delta$, everybody $\omega(\log n)$

Overview Results

Problem		*Upper* $f(n)$	*Upper* $w(n)$	*Lower* $f(n)$
MINIMUM	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\langle \mathcal{O}(1)^\dagger, n^\varepsilon \rangle$ $\langle \mathcal{O}(\log_\Delta n)^\dagger, \mathcal{O}(\Delta + \log_\Delta n)^\dagger \rangle$	$\mathcal{O}(n)$	$\langle \Omega(\log_\Delta n)^\dagger, \Delta \rangle$
		$\langle O(\log_\Delta n \log\log \Delta),$ $O(\Delta + \log_\Delta n \log\log \Delta)\rangle^\ddagger$	$\mathcal{O}(n)$	$\Omega(\log\log n)^\ddagger$
SELECTION	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\left\langle \mathcal{O}(\log\log n)^\dagger, \mathcal{O}\left(\sqrt{n}\right)^\dagger \right\rangle$ $\left\langle \mathcal{O}\left(\frac{\log n}{\log\log n}\right)^\dagger, \mathcal{O}(\log^2 n)^\dagger \right\rangle$	$\mathcal{O}(n)^\dagger$	$\langle \Omega(\log_\Delta n)^\dagger, \Delta \rangle$

$f(n)$ – fragile complexity; $w(n)$ – work; $\langle f_m(n), f_{rem}(n) \rangle$ – fragile complexity for the selected element (minimum/median) and the remaining elements, respectively; † – holds in expectation, ‡ – holds with high probability.

Lower Bound in this Trade-off

Given Δ

$f_m(n) = \Omega(\log_\Delta n)$ (matching the algorithm)

Main Lemma

Minimum Element must be compared to a value

in $(\frac{1}{100\Delta}, 1)$ with constant probability

because indistinguishable from large value on first comparison

... in $(\frac{1}{(100\Delta)^2}, \frac{1}{100\Delta})$ with constant probability

... in $(\frac{1}{(100\Delta)^3}, \frac{1}{(100\Delta)^2})$ with constant probability

...

Linearity of expectation!

Overview Results

Problem		Upper $f(n)$	Upper $w(n)$	Lower $f(n)$
Minimum	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\langle \mathcal{O}(1)^{\dagger}, n^{\varepsilon} \rangle$ $\langle \mathcal{O}(\log_{\Delta} n)^{\dagger}, \mathcal{O}(\Delta + \log_{\Delta} n)^{\dagger} \rangle$	$\mathcal{O}(n)$	$\langle \Omega(\log_{\Delta} n)^{\dagger}, \Delta \rangle$
		$\langle O(\log_{\Delta} n \log\log \Delta),$ $O(\Delta + \log_{\Delta} n \log\log \Delta) \rangle^{\ddagger}$	$\mathcal{O}(n)$	$\Omega(\log\log n)^{\ddagger}$
Selection	Determ.	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\Omega(\log n)$
	Rand.	$\langle \mathcal{O}(\log\log n)^{\dagger}, \mathcal{O}\left(\sqrt{n}\right)^{\dagger} \rangle$ $\left\langle \mathcal{O}\left(\frac{\log n}{\log\log n}\right)^{\dagger}, \mathcal{O}(\log^2 n)^{\dagger} \right\rangle$	$\mathcal{O}(n)^{\dagger}$	$\langle \Omega(\log_{\Delta} n)^{\dagger}, \Delta \rangle$

$f(n)$ – fragile complexity; $w(n)$ – work; $\langle f_m(n), f_{rem}(n) \rangle$ – fragile complexity for the selected element (minimum/median) and the remaining elements, respectively; † – holds in expectation, ‡ – holds with high probability.

Further Results

Randomized Median / Selection

So far more complicated algorithms, almost similar bounds:
Fragile complexity of median is like the minimum, the other elements somewhat worse
Lower Bounds inherited from minimum by padding.

Merge Sort

Randomized Merge sort achieves fragile complexity of $\mathcal{O}(\log n)$

Future Work and Open Questions

- Geometric Problems: Predicates (or something else?)
- Simple deterministic solution for sorting?
- Simpler and improved Median Algorithms!
- Data Oblivious Algorithms?
- Other comparison based problems?