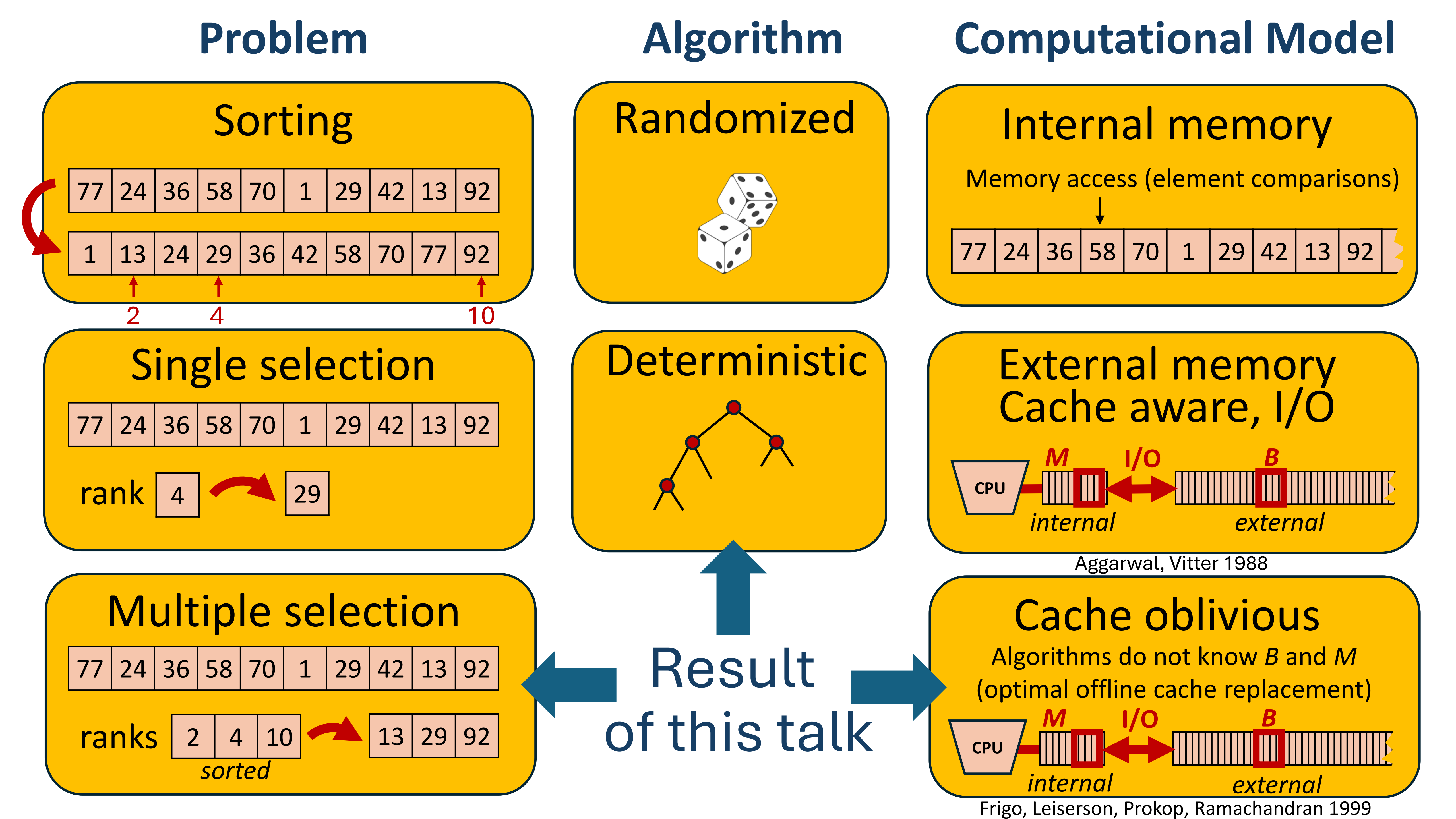
Deterministic Cache-Oblivious Funnelselect

Gerth Stølting Brodal
AARHUS UNIVERSITY

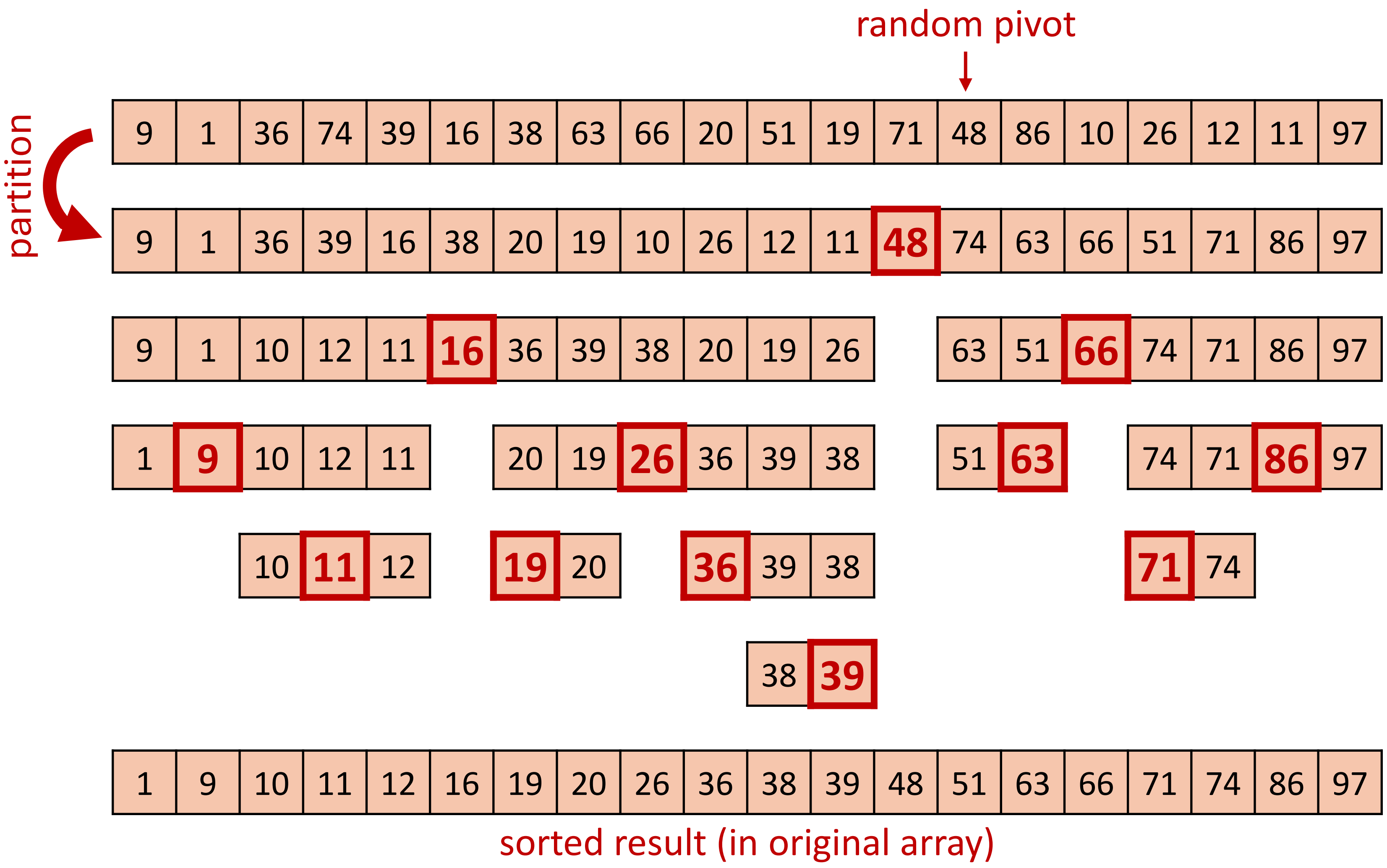
Sebastian Wild
UNIVERSITY OF LIVERPOOL

19th Scandinavian Symposium on Algorithm Theory (SWAT), June 12-14, 2024, Helsinki, Finland

▶ **Theorem 1.** *There exists a deterministic cache-oblivious algorithm solving the multiple-selection problem using* $O(\mathcal{B} + N)$ *comparisons and* $O\big(\mathcal{B}_{\mathrm{I/O}} + \frac{N}{B}\big)$ *I/Os in the worst case, assuming a tall cache* $M \geq B^{1+\varepsilon}$.

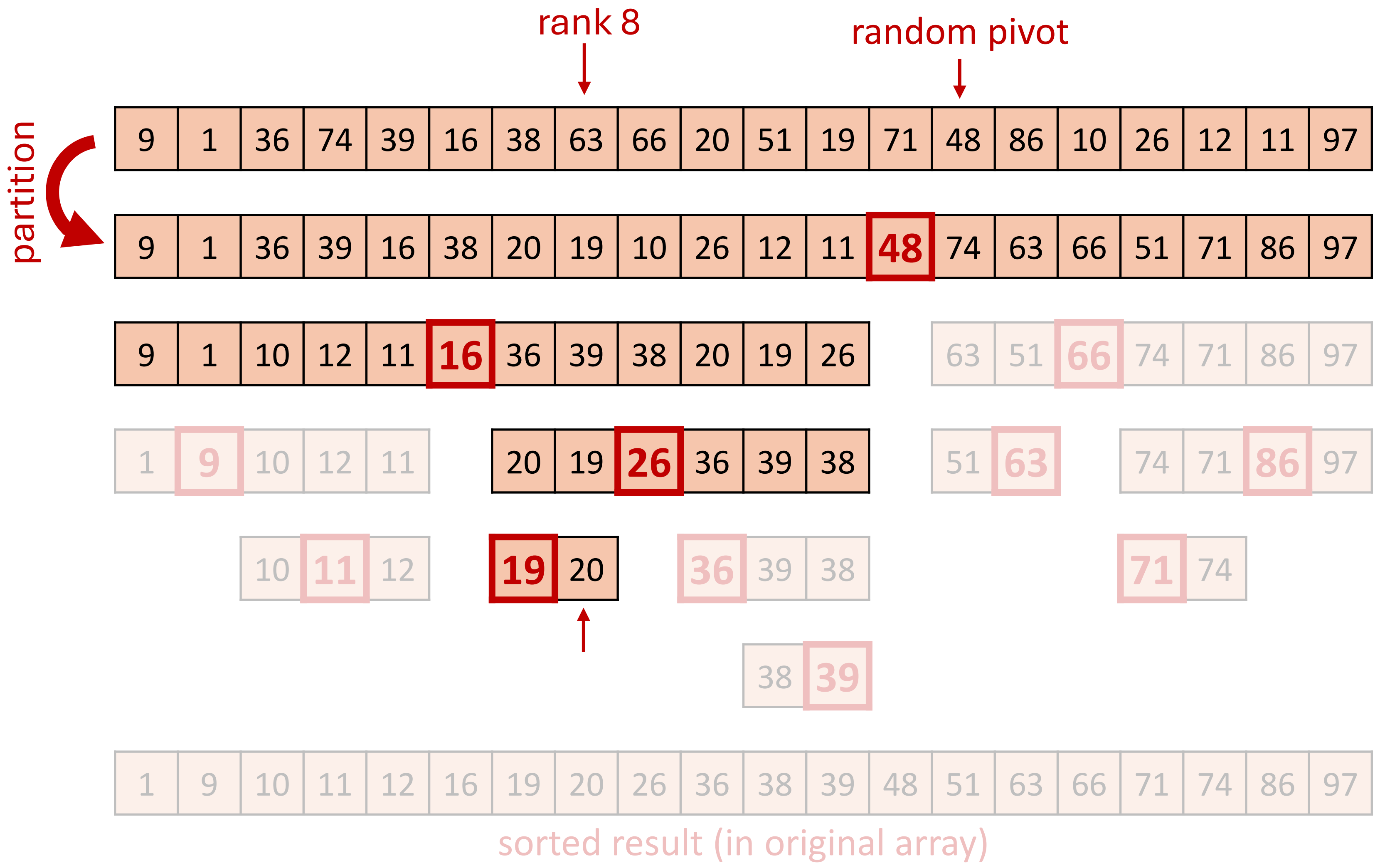

Problem
Algorithm
Computational Model
Sorting
77 24 36 58 70 1 29 42 13 92
1 13 24 29 36 42 58 70 77 92
2
4
10
Single selection
77 24 36 58 70 1 29 42 13 92
rank 4
29
Multiple selection
77 24 36 58 70 1 29 42 13 92
ranks 2 4 10
sorted
13 29 92
Randomized
Deterministic
Result of this talk
Internal memory
Memory access (element comparisons)
77 24 36 58 70 1 29 42 13 92
External memory Cache aware, I/O
CPU
M
I/O
B
internal
external
Aggarwal, Vitter 1988
Cache oblivious
Algorithms do not know B and M
(optimal offline cache replacement)
CPU
M
I/O
B
internal
external
Frigo, Leiserson, Prokop, Ramachandran 1999

Internal Memory **Sorting** – Randomized



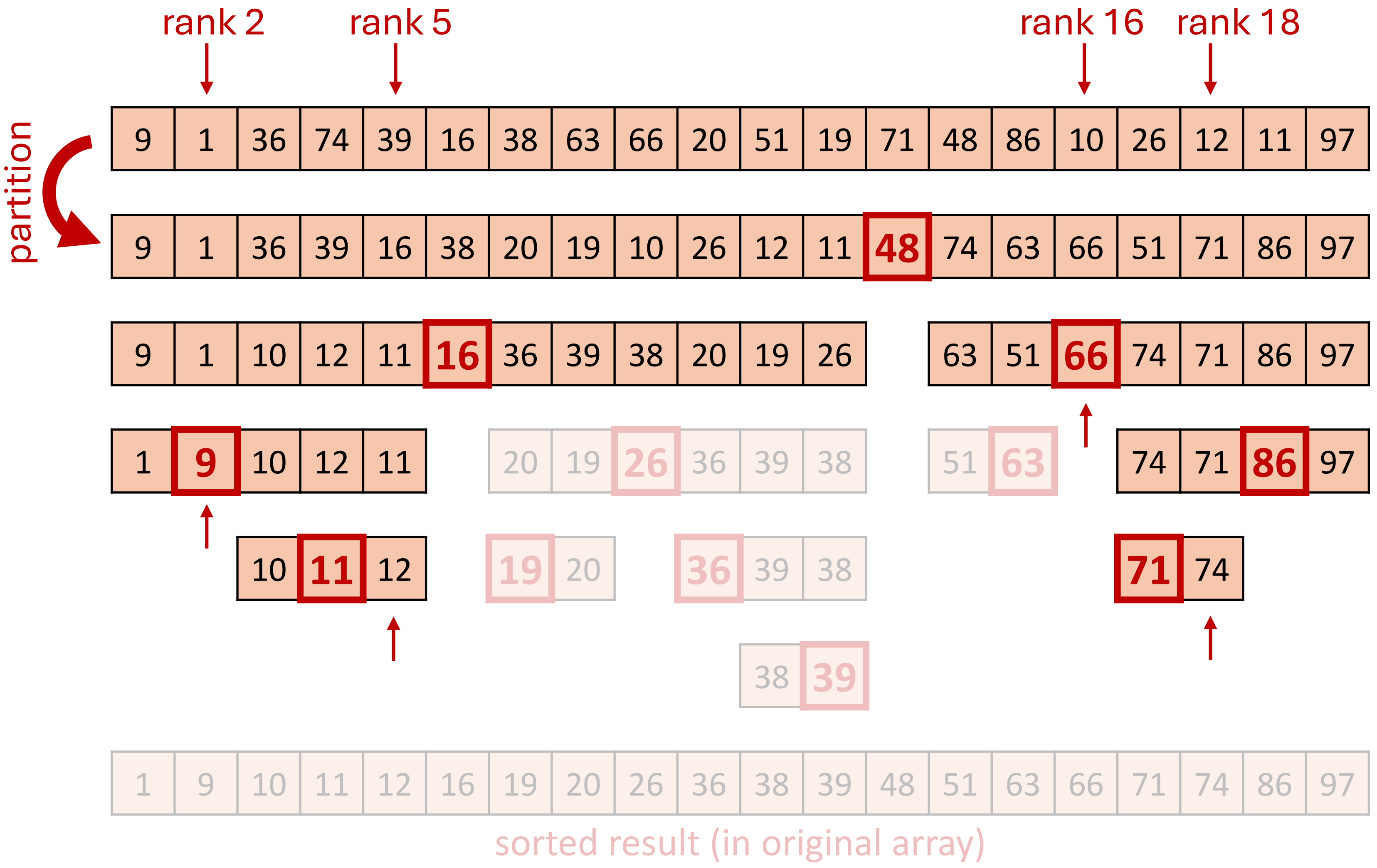
- Quicksort [Hoare 1959]
- Expected time $O(n \log_2 n)$
- Cache-oblivious $O\left(\frac{n}{B} \log_2 \frac{n}{M}\right)$ I/Os

Internal Memory **Single Selection** – Randomized



- Quickselect [Hoare 1961]
- Expected time $O(n)$
- Cache-oblivious $O\left(\frac{n}{B}\right)$ I/Os

Internal Memory **Multiple Selection** – Randomized



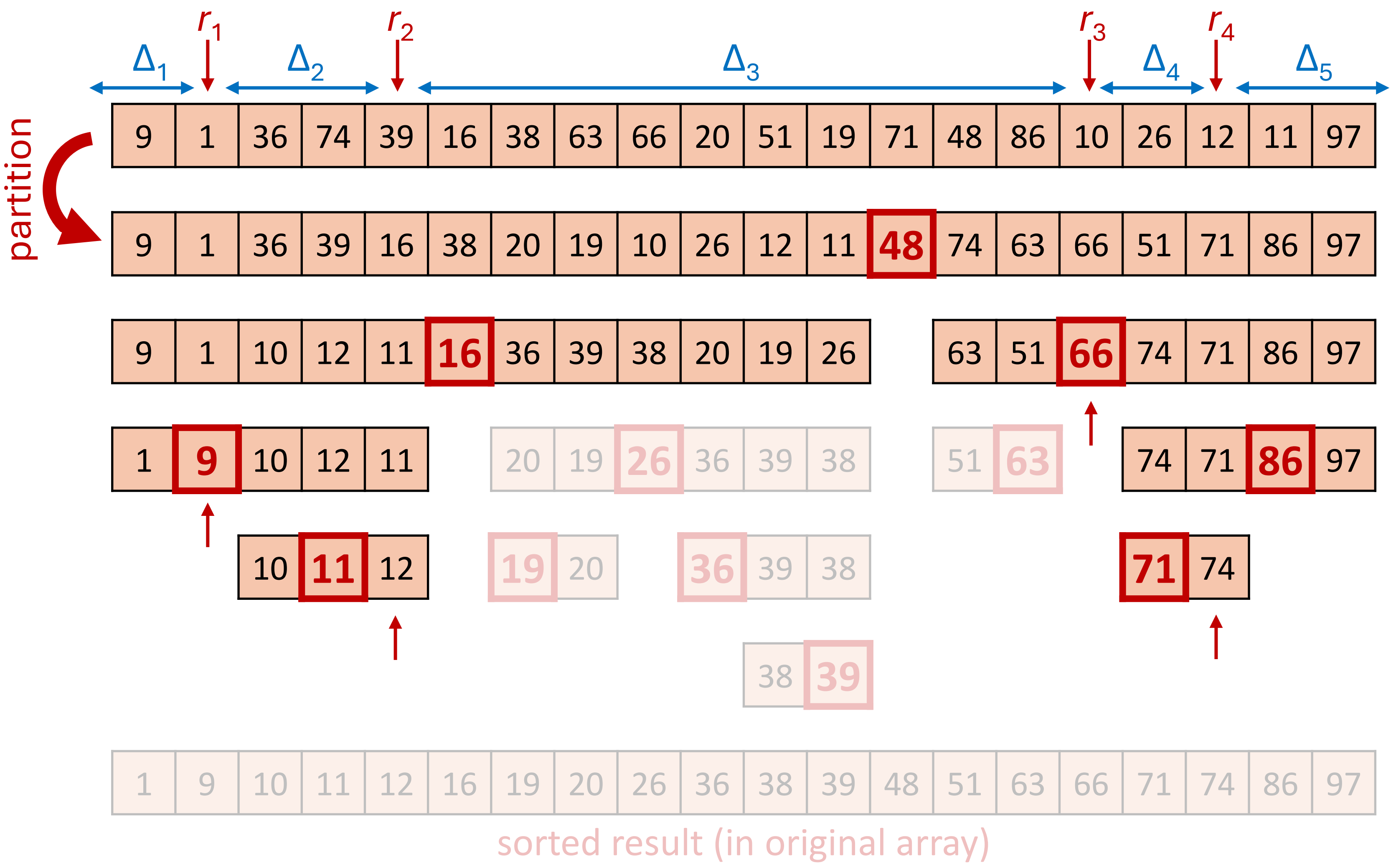
- Multiple selection [Chambers 1971]
- Expected time

  $O(n \log_2 q)$

  q = # query ranks
- Cache-oblivious

  $O\left(\frac{n}{B} \log_2 q\right)$ I/Os

Internal Memory **Multiple Selection** – Randomized



- Multiple selection [Chambers 1971]
- Expected time [Prodinger 1995]

$O(\mathcal{B} + n)$

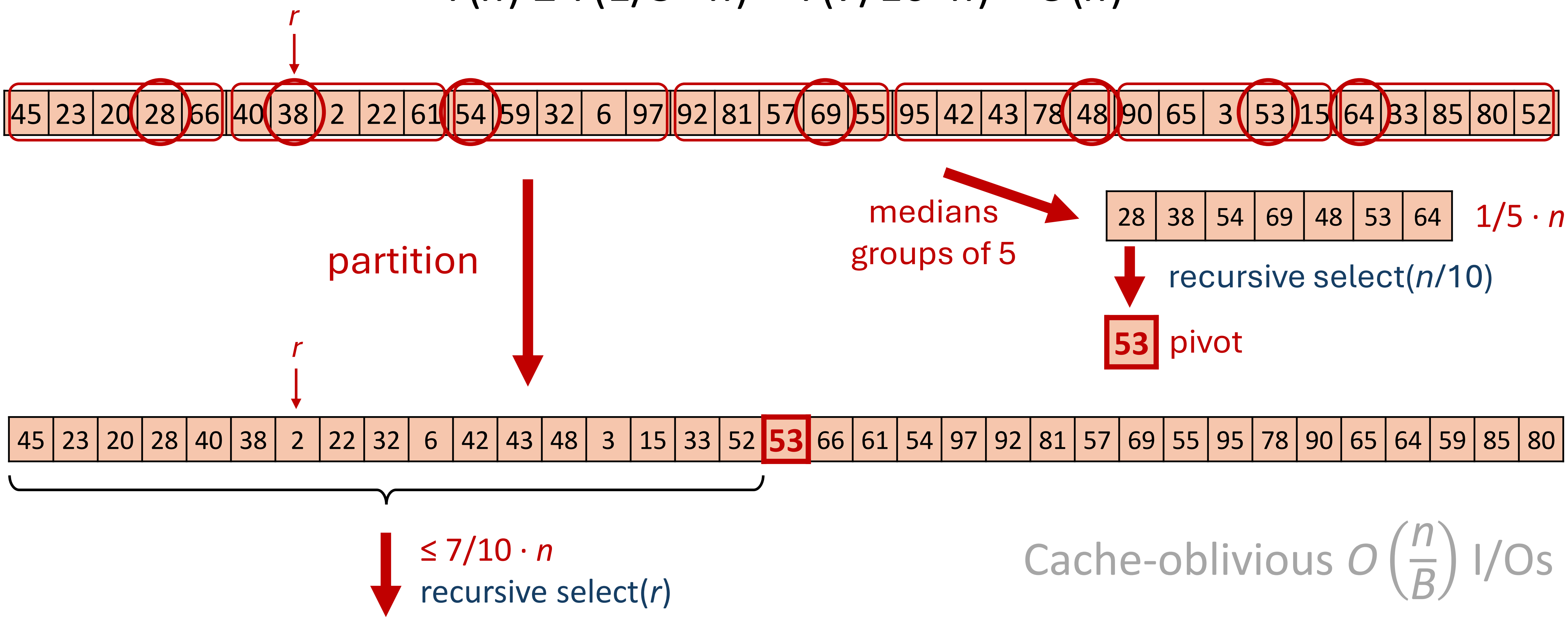
$$\mathcal{B} = \sum_{i=1}^{q+1} \Delta_i \log_2 \frac{n}{\Delta_i}$$

(query rank entropy)

Internal Memory **Single Selection** – Deterministic

- [Blum, Floyd, Pratt, Rivest, Tarjan 1973] Worst-case linear time

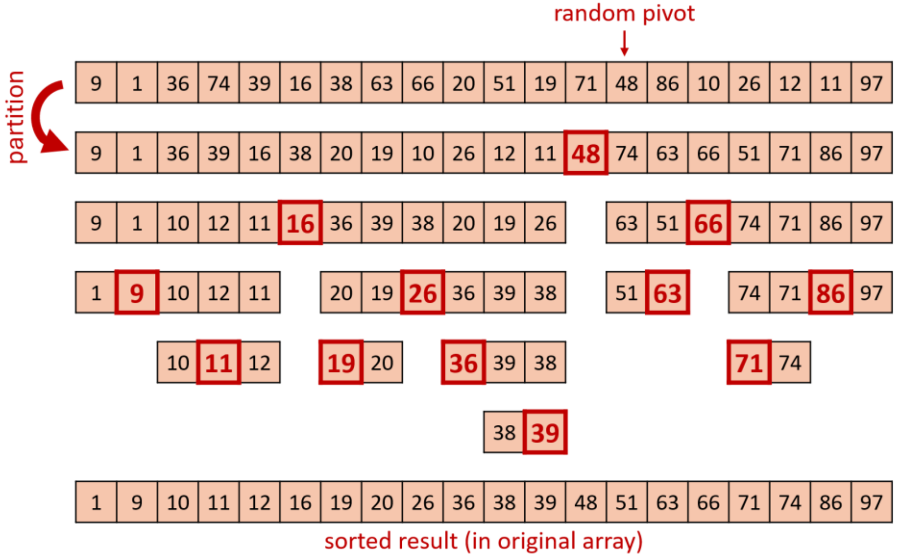
$$T(n) \leq T(1/5 \cdot n) + T(7/10 \cdot n) + O(n)$$



Cache-oblivious $O\left(\frac{n}{B}\right)$ I/Os

Internal Memory **Distribution Sorting** – Deterministic

Quicksort
[Hoare 1959]



+

Linear time median → pivots
[Blum, Floyd, Pratt, Rivest, Tarjan 1973]

↓

Deterministic distribution sorting

optimal $O(n \log_2 n)$ time

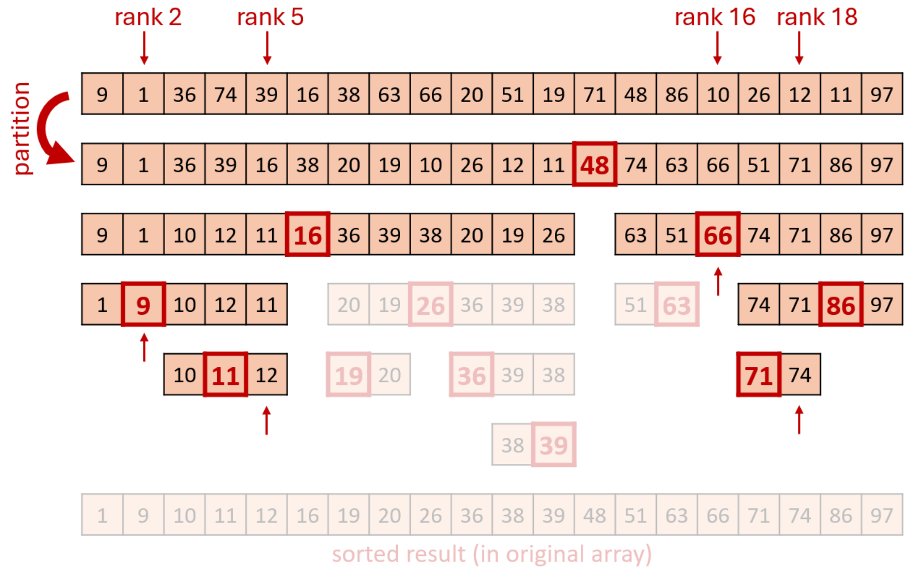
Cache-oblivious $O\left(\frac{n}{B}\log_2\frac{n}{M}\right)$ I/Os

Internal Memory **Multiple Selection** – Deterministic

Multiple selection
[Chambers 1971]

+

Linear time median → pivots
[Blum, Floyd, Pratt, Rivest, Tarjan 1973]



↓

Deterministic multiple selection
[Dobkin, Munro 1981]

optimal $O(\mathcal{B} + n)$ time

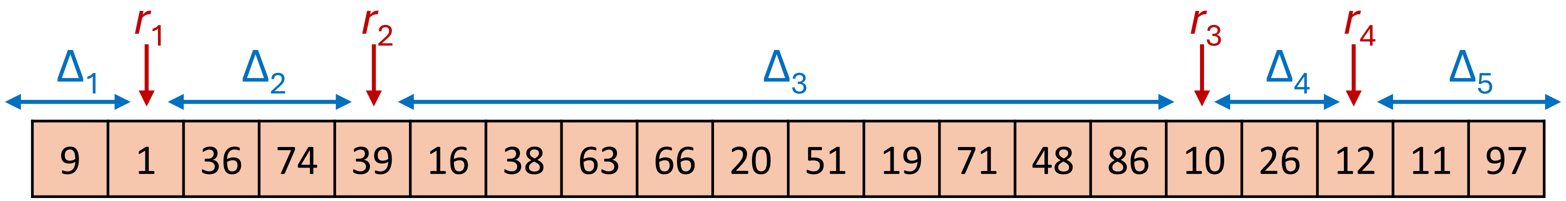
Cache-oblivious $O((\mathcal{B} + n) / B)$ I/Os

Summary Internal Memory

	Randomized / Deterministic	Cache-oblivious I/Os	Optimal I/O ?
Sorting	$O(n \log_2 n)$	$O\left(\frac{n}{B} \log_2 \frac{n}{M}\right)$	$O\left(\frac{n}{B} \log_{M/B} \frac{n}{M}\right)$
Single selection	$O(n)$	$O(n / B)$	✓
Multiple selection	$O(\mathcal{B} + n)$	$O(n / B + \mathcal{B} / B)$	$O(n / B + \mathcal{B} / B / \log_2 \frac{M}{B})$

$$\mathcal{B} = \sum_{i=1}^{q+1} \Delta_i \log_2 \frac{n}{\Delta_i}$$

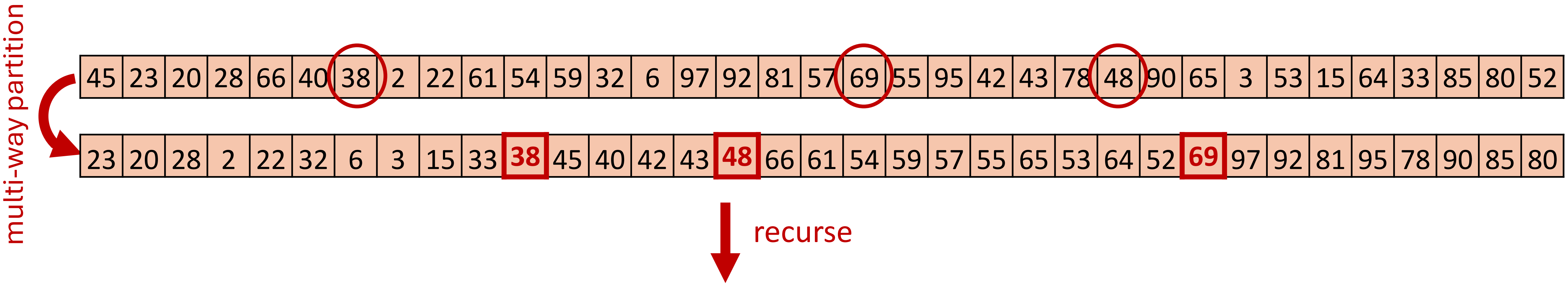
(query rank entropy)



Lower bound
[Dobkin, Munro 1981] +
[Arge, Knudsen, Larsen 1993]

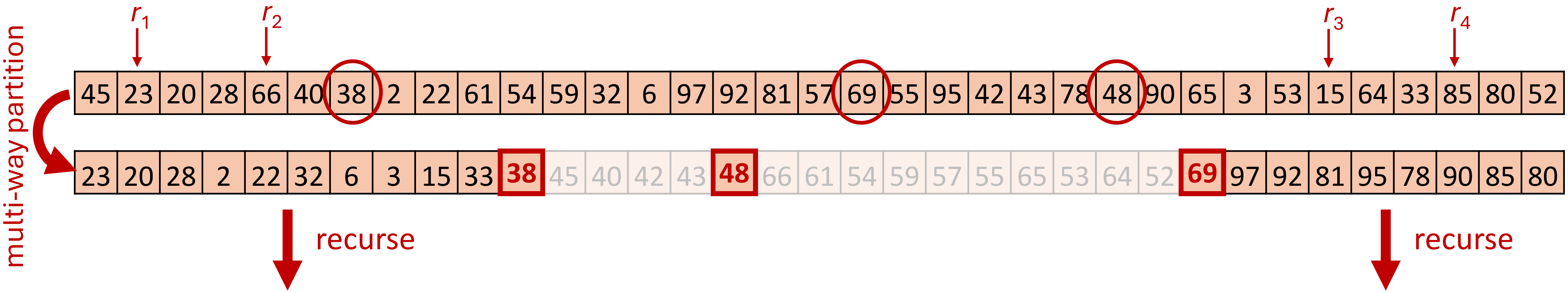
Upper bounds
Randomized [ESA 2023]
Deterministic [SWAT 2024]

Cache-aware **Distribution Sort** – Randomized



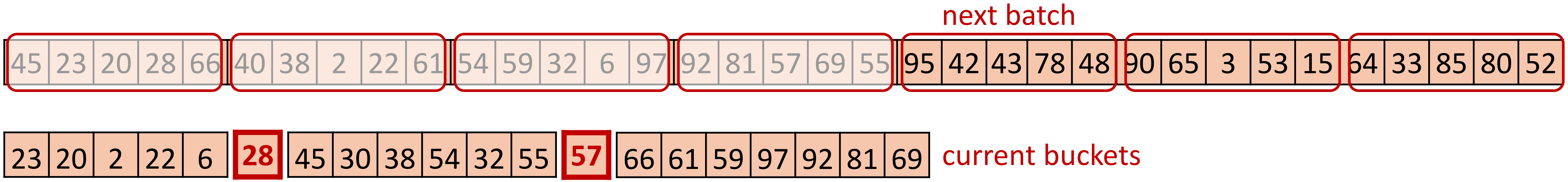
- Generalize Quicksort
- Pick $\Theta(M/B)$ random pivots
- Multi-way partition: Scan and distribute to buckets
  [internal memory: $\Theta(M/B)$ pivots + 1 block input + 1 block per output bucket]
- Recurse on buckets ($\leq M$ elements sorted in internal memory)
- Expected $O\left(\frac{n}{B}\log_{M/B}\frac{n}{M}\right)$ I/Os and $O(n \log n)$ internal work

Cache-aware **Multiple Selection** – Randomized



- $\Theta(M/B)$-way distribution sort with pruning (à la Chambers)
- Expected optimal $O(n / B + \mathcal{B} / B / \log_2 \frac{M}{B})$ I/Os

Cache-aware **Multi-way Partition** – Deterministic



- **Goal** : create k = $\Theta(M/B)$ buckets each of size $O(n/k)$
- Repeatedly distribute batches of n/k elements into buckets (initially one)
- Split overflowing buckets (> $2 \cdot n/k$ elements; at most $3n/k$)
  [using Blum *et al*. median finding algorithm; split creates pivot]
- Result : ≤ k buckets each of size $[n/k, 2n/k]$
- **Distribution sort** : $O\left(\frac{n}{B} \log_{M/B} \frac{n}{M}\right)$ I/Os and $O(n \log n)$ internal work
- **Multiple selection** : $O(n / B + \mathcal{B} / B / \log_2 \frac{M}{B})$ I/Os

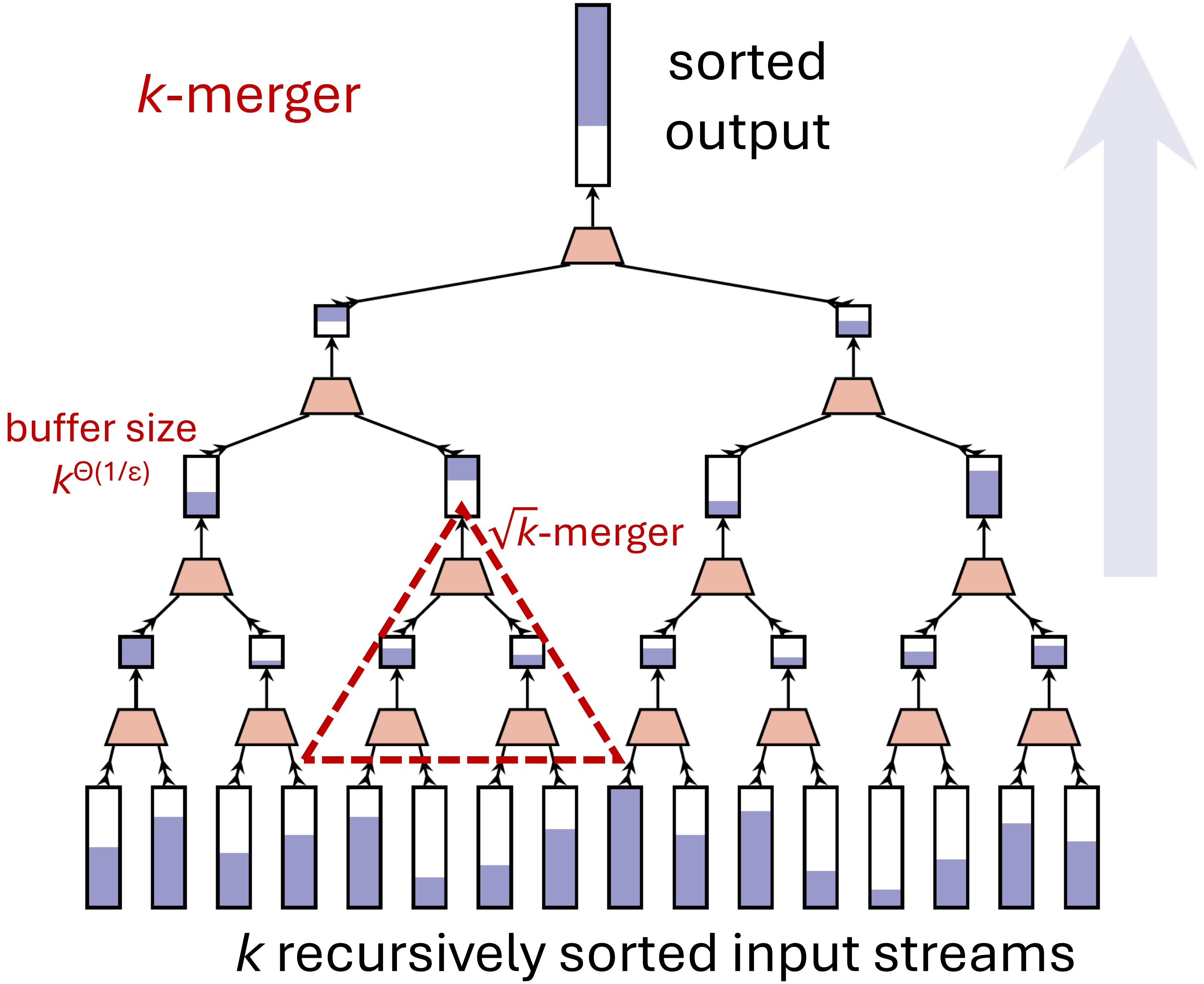
Summary Cache-aware

Sorting	$O\left(\frac{n}{B}\log_{M/B}\frac{n}{M}\right)$ I/Os
Single selection	$O(n / B)$ I/Os
Multiple selection	$O(n / B + \mathcal{B} / B / \log_2 \frac{M}{B})$ I/Os

[Hu, Tao, Yang, Zhou 2014] achieved both I/O and work optimality

[Frigo, Leiserson, Prokop, Ramachandran 1999]

Cache-oblivious **Funnelsort** – Deterministic



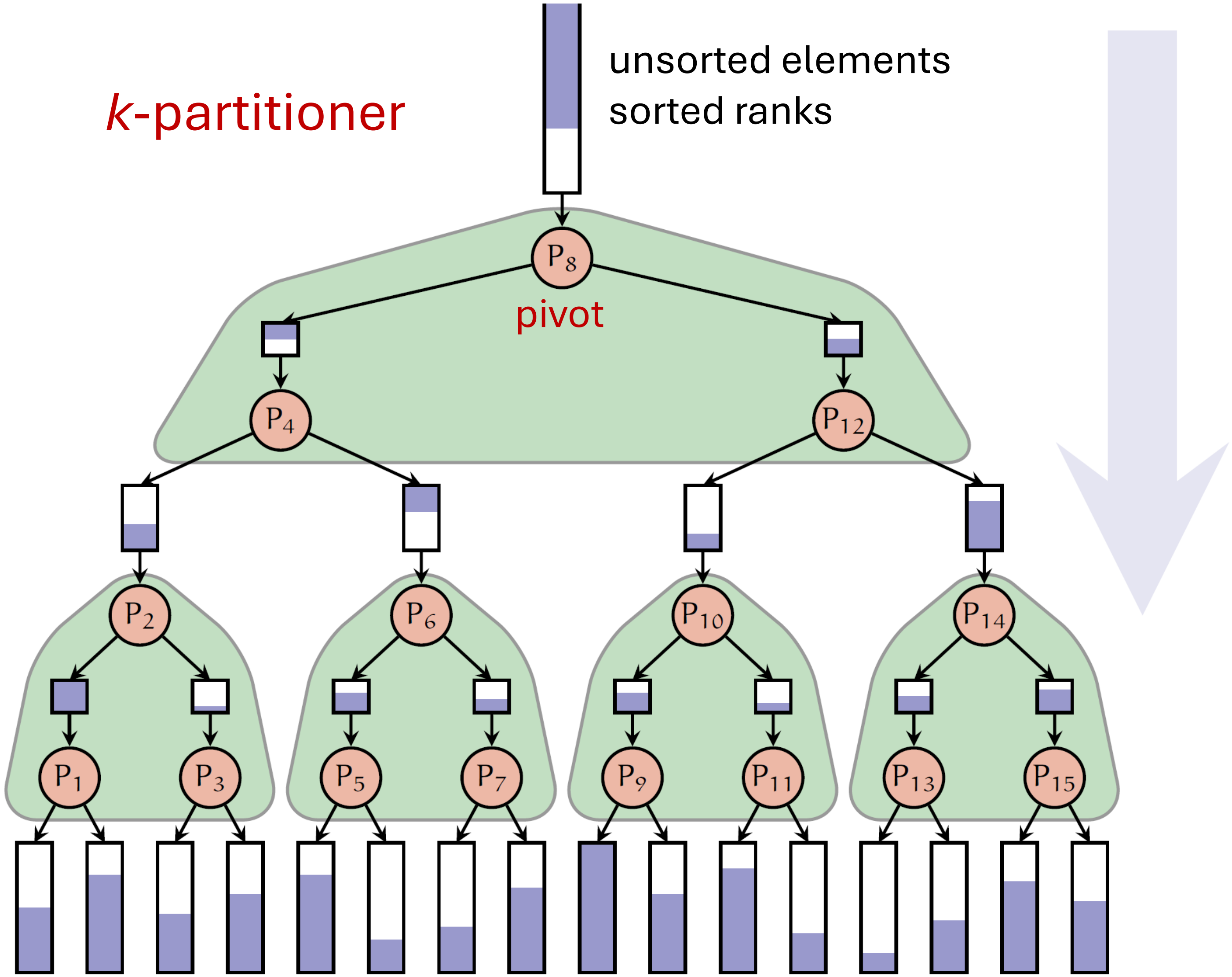
- Tall cache assumption

  $$M \geq B^{1+\varepsilon}$$

  Neccessary [Brodal, Fagerberg 2002]

- Binary mergesort with buffered output
- *k*-merger , $k = n^{\Theta(\varepsilon)}$
- $O\left(\frac{n}{B}\log_{M/B}\frac{n}{M}\right)$ I/Os

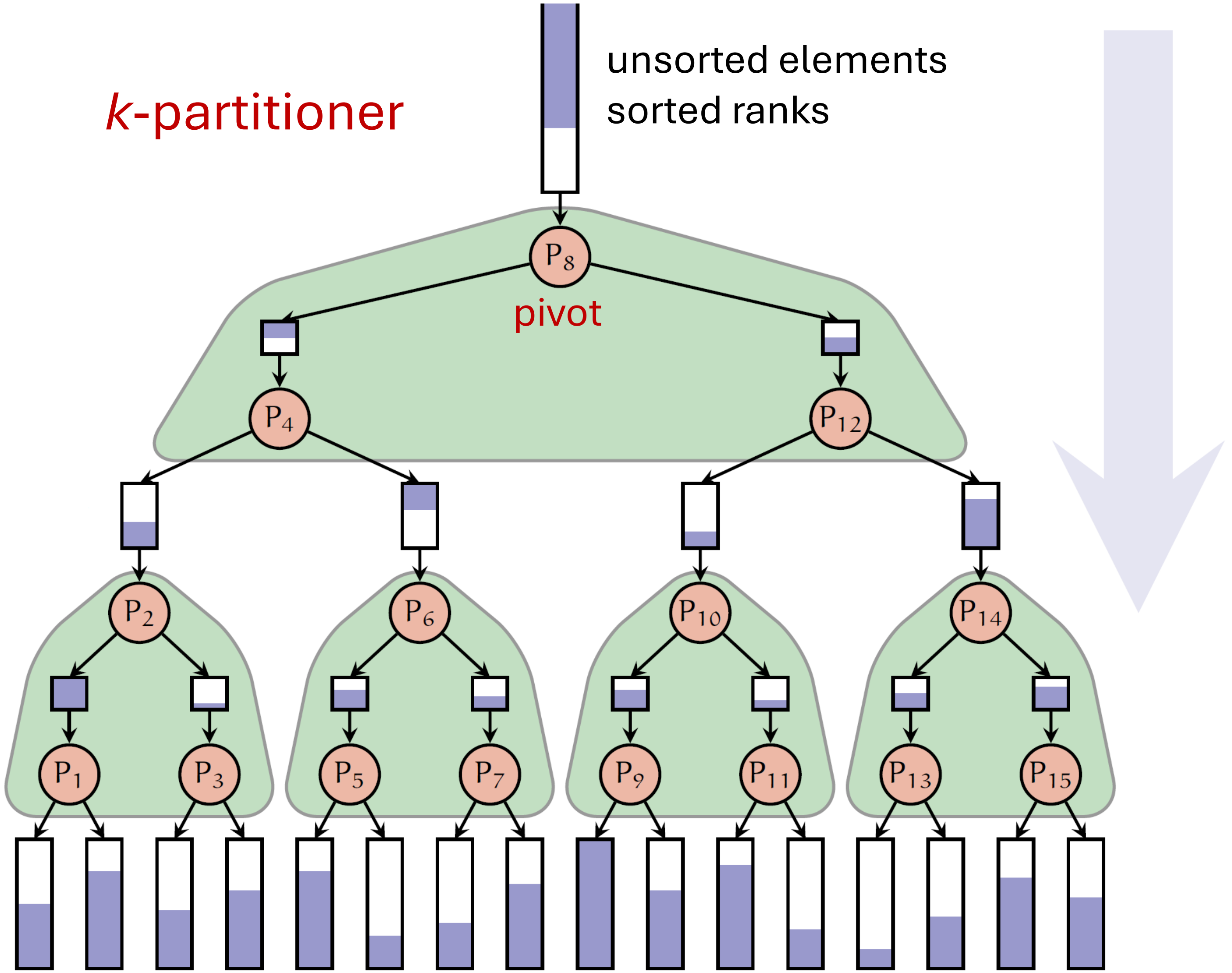
Cache-oblivious **Multiple Selection** – Idea



- Reverse computation of *k*-merger → *k*-partitioner
- algorithm à la Chambers

- Challenges
  - Pivots ?
  - Pruning subtree computations before knowning ranks of pivots ?

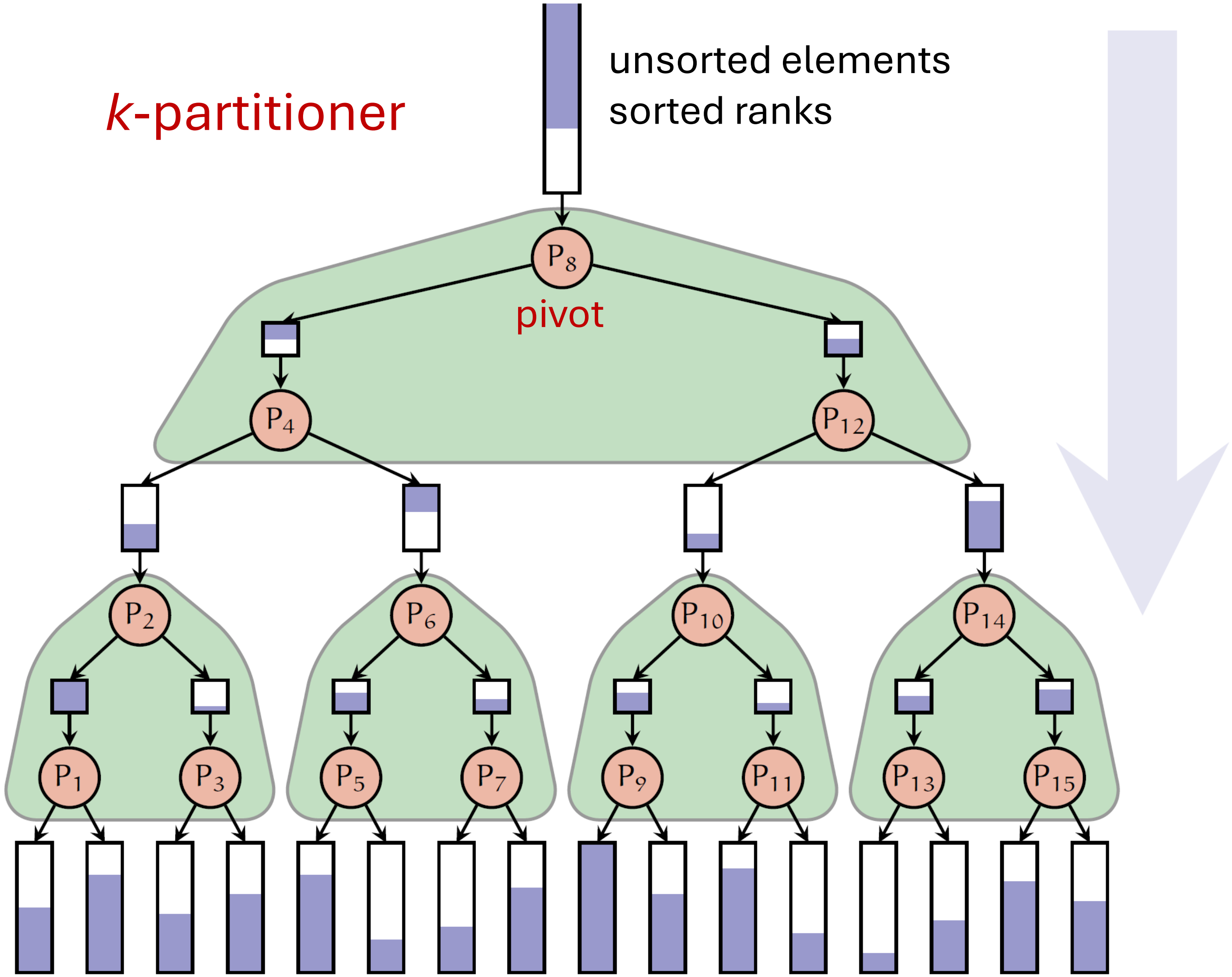
Cache-oblivious **Multiple Selection** – Randomized



- Sort n / log n size sample
- Select $k = n^{\Theta(\varepsilon)}$ pivots uniformly in sample
- Estimate pivot ranks within $\pm n^{2/3}$ w.h.p.
- Prune inside k-partitioner subtrees w.h.p. no query
- Buckets just sort
- Expected $O(n / B + \mathcal{B} / B / \log_2 \frac{M}{B})$ I/Os

[Brodal, Wild 2024]

Cache-oblivious **Multiple Selection** – Deterministic

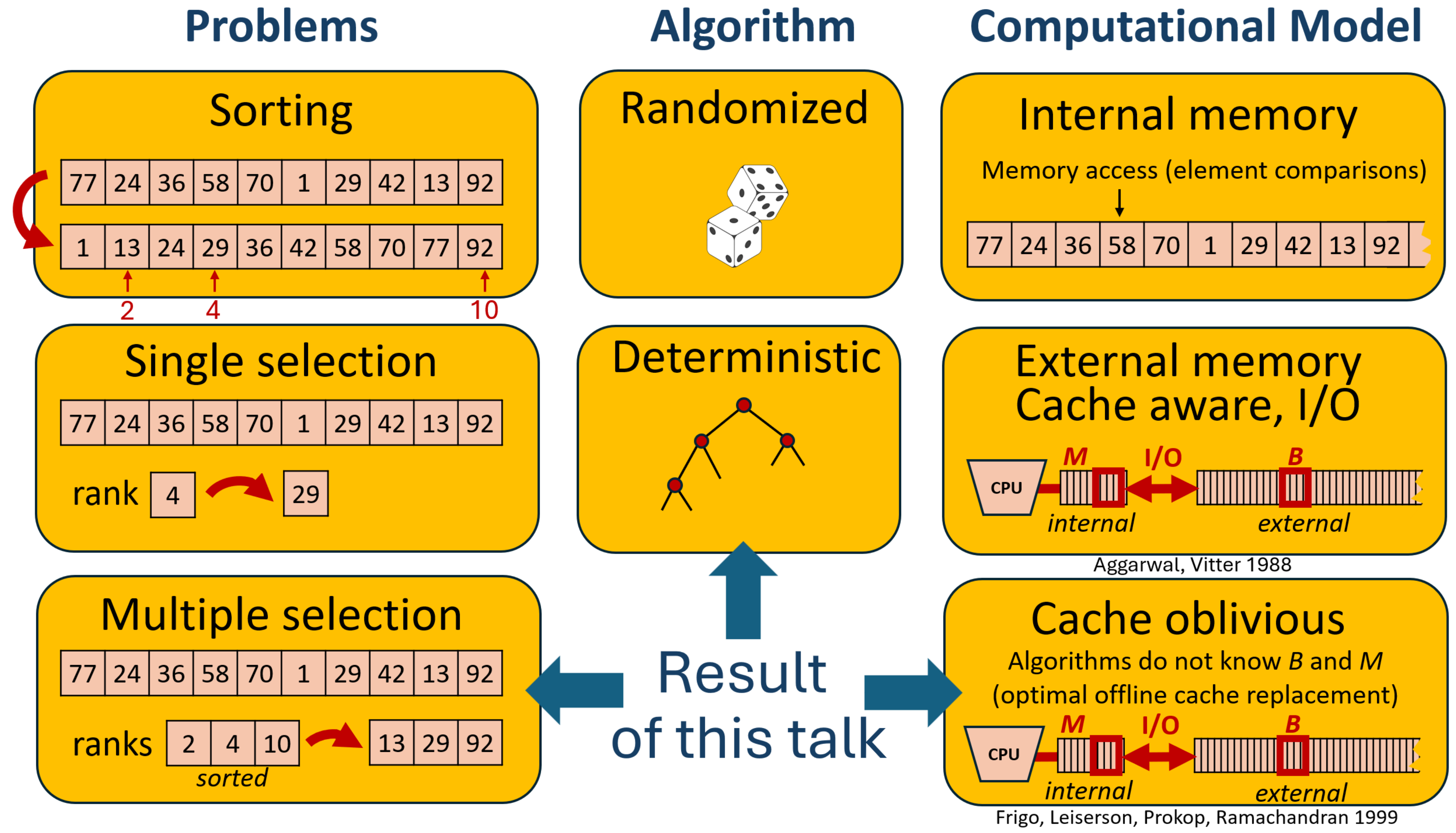


- Entropy bound ≈ sort → sort
- Smaller k
- Prune buckets with no query
  - $\geq n/2$ elements pruned
- Recurse on buckets
- Pivots deterministically
  - Incremental batches size n/k
  - Split bucket + rebuild k-partitioner
- $O(n / B + \mathcal{B} / B / \log_2 \frac{M}{B})$ I/Os

$$k = n/\Delta$$

$$\Delta = \min \{\Delta_i \mid \sum_{\Delta_j \leq \Delta_i} \Delta_j \geq n/2\}$$

Summary


Problems
Algorithm
Computational Model
Sorting
77 24 36 58 70 1 29 42 13 92
1 13 24 29 36 42 58 70 77 92
2 4 10
Randomized
Internal memory
Memory access (element comparisons)
77 24 36 58 70 1 29 42 13 92
Single selection
77 24 36 58 70 1 29 42 13 92
rank 4 29
Deterministic
External memory
Cache aware, I/O
CPU
M
I/O
B
internal
external
Aggarwal, Vitter 1988
Multiple selection
77 24 36 58 70 1 29 42 13 92
ranks 2 4 10
sorted
13 29 92
Result
of this talk
Cache oblivious
Algorithms do not know B and M
(optimal offline cache replacement)
CPU
M
I/O
B
internal
external
Frigo, Leiserson, Prokop, Ramachandran 1999