MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

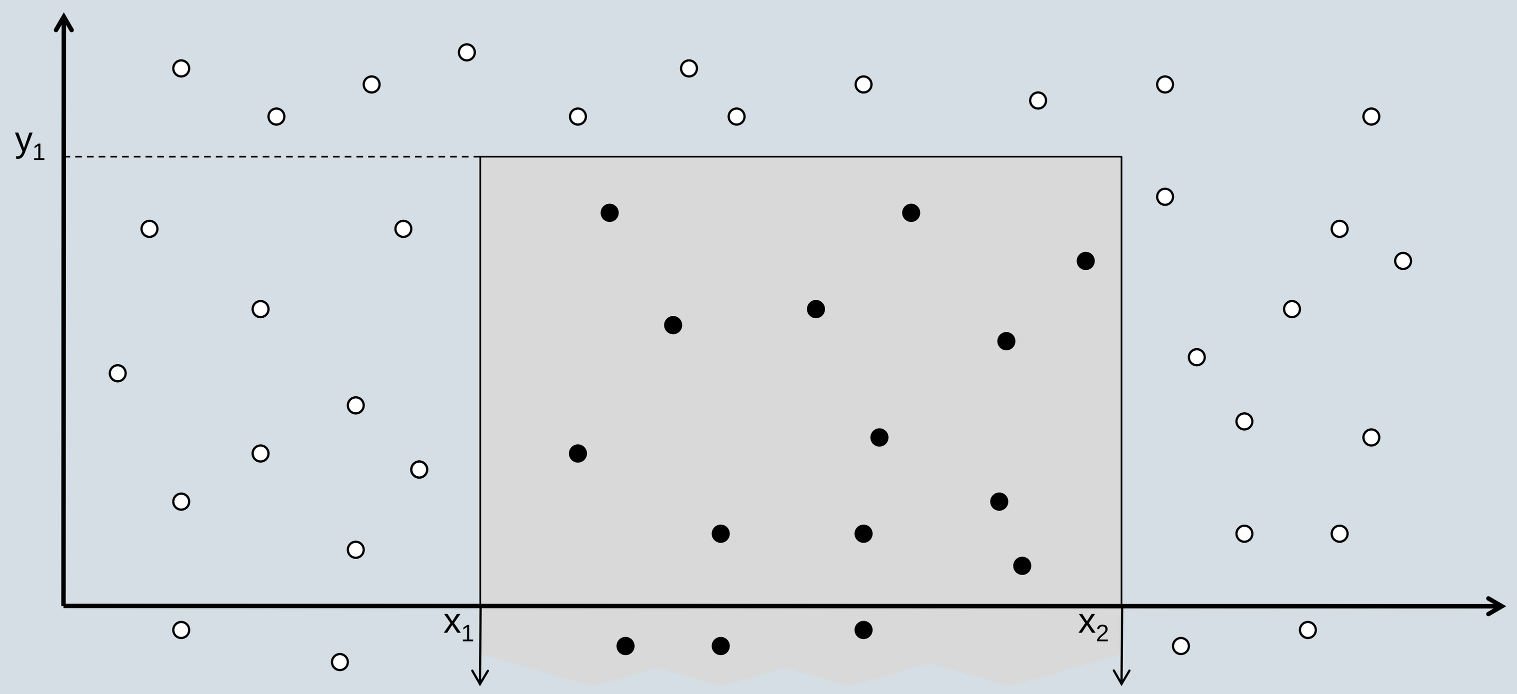
Konstantinos Tsakalidis
Aarhus University

Orthogonal Planar 3-sided Range Reporting

Store n points (x,y) in the 2-dimensional plane, such as to **report** all t points that lie within any query region of the form $[x_1, x_2] \times (-\infty, y_1]$, namely a **three-sided rectangle** with one side unbounded.

The dynamic setting allows for **insertions** and **deletions** of points (updates).

Efficiency (Complexity) is measured in terms of **(S:** used space**, Q:** query time**, U:** update time**)**.

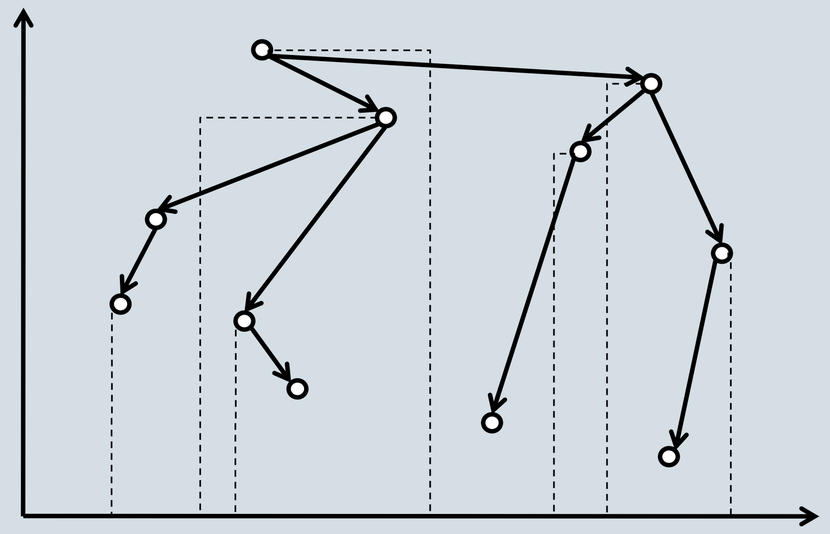


This is a basic problem in computational geometry and finds applications in spatial databases, computer graphics, geographic information systems, and more areas.

Internal Memory

Pointer Machine

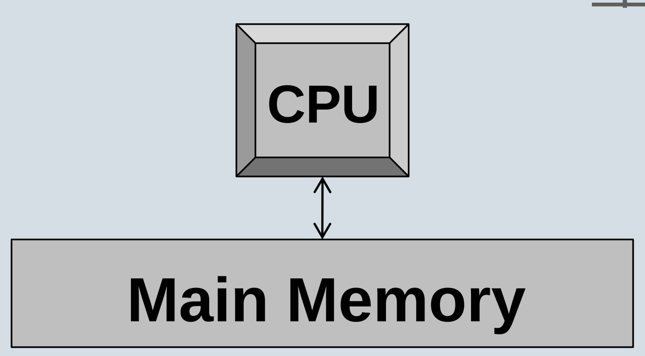
Data resides in records (nodes) that can be accessed via pointers (links).



The **priority search tree** achieves $(n, \log n + t, \log n)$.
[McCreight, SIAM J.Comp.'85]

Random Access Machine

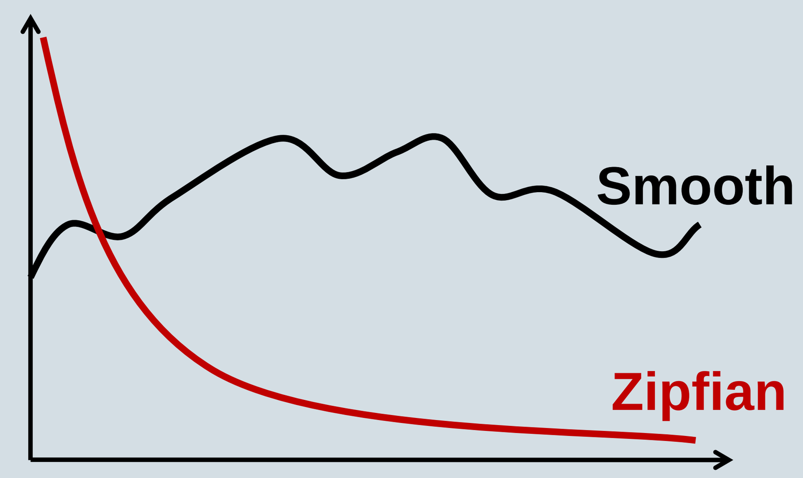
Data is stored in an infinite array of cells, each containing a word of bits. A CPU can access any cell and perform arithmetic or bitwise operations.



The **fusion tree** achieves $(n, \frac{\log n}{\log\log n} + t, \frac{\log n}{\log\log n})$ or $(n, \sqrt{\log n} + t, \sqrt{\log n})$ where Q and U are expected bounds [Willard, SODA'92].
[Mortensen, SODA'03] reduces U achieving $(n, \frac{\log n}{\log\log n} + t, \log^{\omega} n), \omega < 1$.

Probabilistic Distributions and Our Results

Q and U can be significantly reduced if we assume that the coordinates are being drawn from an unknown, continuous μ-random distribution.



Two examples are the Zipfian distribution, which is commonly used in practice, and the Smooth distribution, which is a generalization of many known distributions.

We attain $(n, \log n + t, \log\log n)$ when both x- and y-coordinates are μ-random, and $(n, \log\log n + t, \log\log n)$ when furthermore x is zipfian. All reduced complexities are expected with high probability [K.Tsakalidis et al., ICDT'10].
The latter bounds hold also when x is smooth. U becomes expected amortized [K.Tsakalidis, Brodal et al., ISAAC'09].

Accordingly, in the I/O model, we attain $(\frac{n}{B}, \log_B n + \frac{t}{B}, \log_B \log n)$ when both x and y are μ-random. When x is zipfian and y is smooth, then $(\frac{n}{B}, \log_B \log n + \frac{t}{B}, \log_B \log n)$ can be attained. All reduced complexities are expected with high probability.
[K.Tsakalidis et al., ICDT'10]
$(\frac{n}{B}, \log\log_B n + \frac{t}{B}, \log_B \log n)$ can be achieved by only assuming that x is smooth. Here, U is expected amortized.
[K.Tsakalidis, Brodal et al., ISAAC'09]

Static Setting

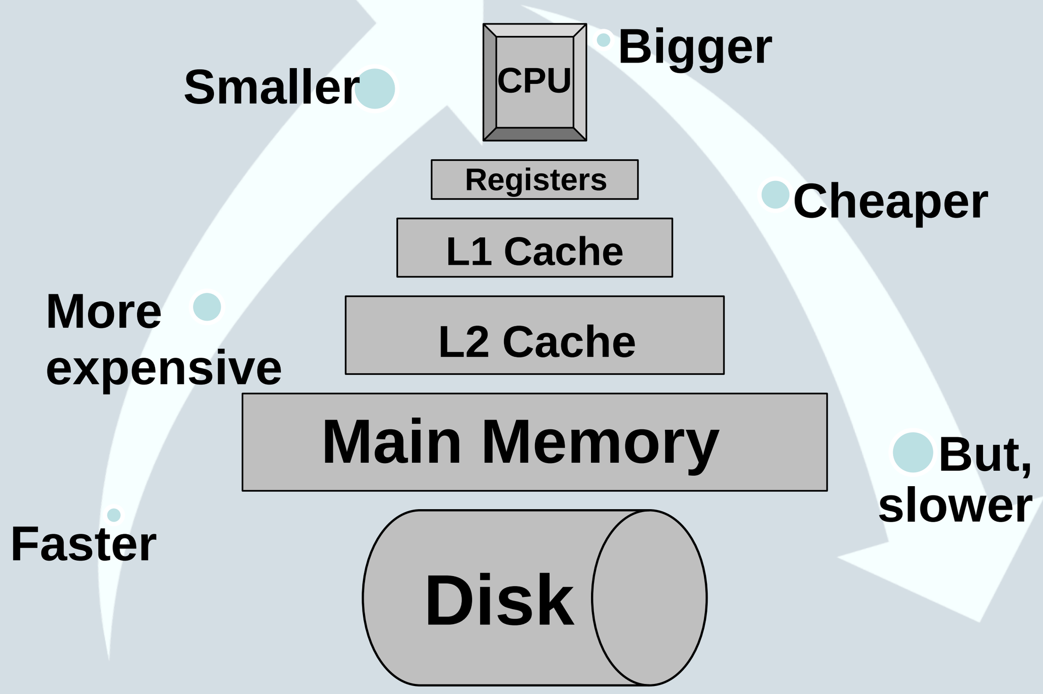
In the pointer machine $(n, \log\log n + t, -)$ is possible, when no updates are allowed [Mehlhorn, A. Tsakalidis et al., IPL'87].

In the RAM, when the x-coordinates are integers in the range [N] (rank space), $(N + n, 1 + t, -)$ is possible.
When both x- and y-coordinates are integers one can achieve $(n, \log\log N + t, -)$ [Brodal et al., FOCS'00].

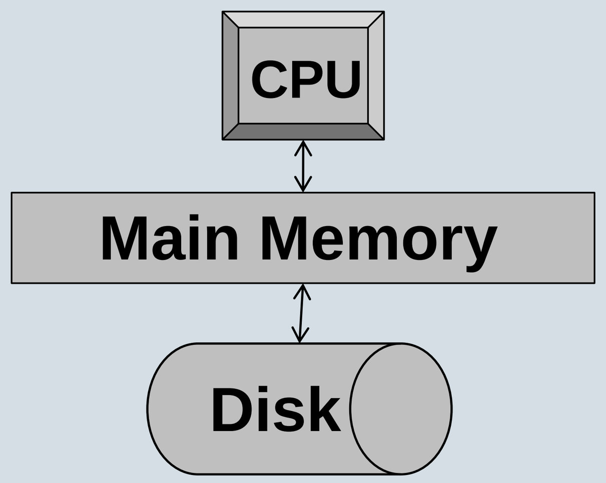
External Memory

I/O Model

The aforementioned models are insufficient to capture the complexity of algorithms running on modern computers. In fact, modern architectures consist of a memory hierarchy, where the bottleneck lies on the transfer of data between consecutive levels, rather than on computation itself.



The Input/Output Model (I/O model) studies two consecutive levels of the memory hierarchy. Data resides in an external memory that consists of blocks of size B. An I/O operation transfers a block to the internal memory of size M, where computation is performed for free.
[Aggarwal, Vitter, C.ACM'88]



The **external priority search tree** is the most efficient data structure for this model and attains $(\frac{n}{B}, \log_B n + \frac{t}{B}, \log_B n)$ [Arge et al., PODS'99].

Cache-oblivious Model

The cache-oblivious model differs from the I/O model only in the fact that the algorithm does not "know" the values of B and M. Thus, it suffices just to study two consecutive levels of the hierarchy, in order to deduce results that hold for all the levels.

There exists a static data structure that attains $(n \log n, \log_B n + \frac{t}{B}, -)$. In a companion paper, the same authors showed that $\Omega(n \log^{\epsilon} \log n)$ space is needed, in order to answer queries in the above query time.
[Afshani, Zeh et al., SoCG'09]

MADALGO – Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation