MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS


Thomas Mølhave
Aarhus University

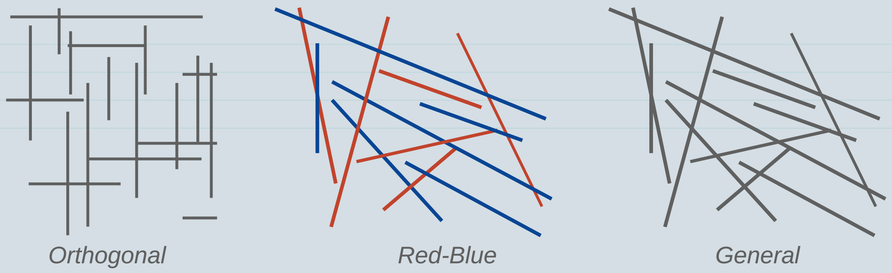
Cache-Oblivious Red-Blue Line Segment Intersection

Motivation

Line Segment intersection

- 2D line segment intersection problems are well studied topics in computational geometry.
- Three common problem types:
  - *General* : Find all intersections between segments in a set
  - *Red-Blue* : Given two sets of line segments, the red set and the blue set, find all intersections between a red segment and a blue segment.

    Common assumption: No two segments of the samme color intersect.
  - *Orthogonal* : Given a set of axis-parallel line segments, find all intersections.



Orthogonal *Red-Blue* *General*

Contribution

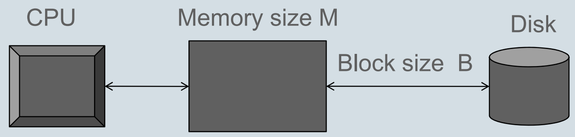
- This work solves the Red-Blue line segment intersection problem in the cache-oblivious model.
- The table below shows what's been done previously for the three problems above. Note that all of the given bounds are optimal, except for the general external memory intersection problem. Our solution is highlighted in red.

	Internal Memory	External Memory	Cache-Oblivious
Orthogonal	$O(N \log N + T)$	$O\left(\frac{N}{B}\log_{\frac{M}{B}}\frac{N}{B}+\frac{T}{B}\right)$	$O\left(\frac{N}{B}\log_{\frac{M}{B}}\frac{N}{B}+\frac{T}{B}\right)$
Red-Blue	$O(N \log N + T)$	$O\left(\frac{N}{B}\log_{\frac{M}{B}}\frac{N}{B}+\frac{T}{B}\right)$	$O\left(\frac{N}{B}\log_{\frac{M}{B}}\frac{N}{B}+\frac{T}{B}\right)$
General	$O(N \log N + T)$	$O\left(\left(\frac{N}{B}+\frac{T}{B}\right)\log_{\frac{M}{B}}\frac{N}{B}\right)$	Open

Cache-Oblivious Model

External memory model

- CPU: Computation. Can only access elements in memory.
- Memory: Finite, holds M elements.
- Disk: Elements transferred to/from disk in block of size B.

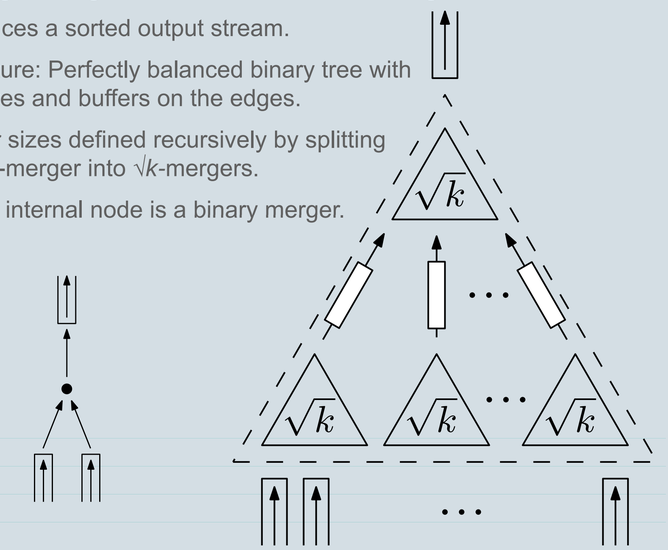


Cache-oblivious model

- In the cache-oblivious model algorithms are designed in the RAM model but *analyzed* in the external memory model.
- Result: Algorithms that are efficient on *all* levels on *any* memory hierarchy .

The *k*-Merger

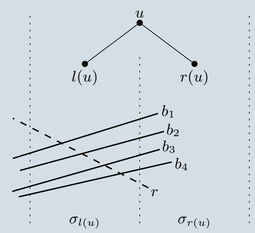
- *k*-merger: merges *k* sorted streams of total length at least k^2.
- Produces a sorted output stream.
- Structure: Perfectly balanced binary tree with *k* leaves and buffers on the edges.
- Buffer sizes defined recursively by splitting The *k*-merger into $\sqrt{k}$-mergers.
- Each internal node is a binary merger.



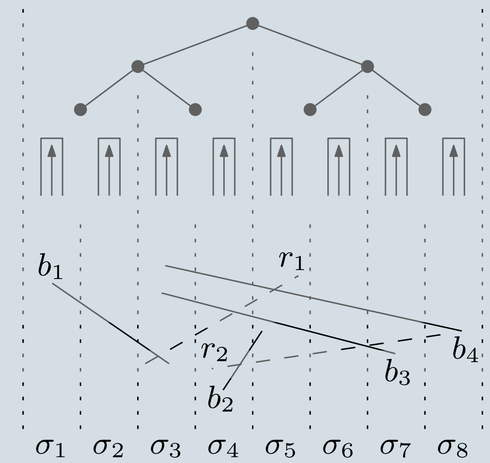
Algorithm Overview

Algorithm based on divide and conquer technique:

1. Split plane into √N slabs with √N endpoints in each slab.
2. Recurse on each slab to find intersections between segments with at least one endpoint in the slab.
3. Use *k*-merger multiple times to find:
   - 3.1. Intersections between short and long segments of different colors.
   - 3.2. Intersections between long red and long blue segments.



Performing steps 3.1 and 3.2 requires significant changes to the binary mergers of the *k*-merger.



References

[1] L. Arge, T. Mølhave, N. Zeh, Cache-Oblivious Red-Blue Line Segment Intersection. In *Proc. ESA*, pp. 88-99, 2008.

[2] G.S. Brodal, R. Fagerberg. Cache oblivious distribution sweeping. In *Proc. ICALP*, pp. 426–438, 2002.

[3] L. Arge, D.E. Vengroff, J.S. Vitter. External-memory algorithms for processing line segments in geographic information systems. *Algorithmica*, 47:1–25, 2007.