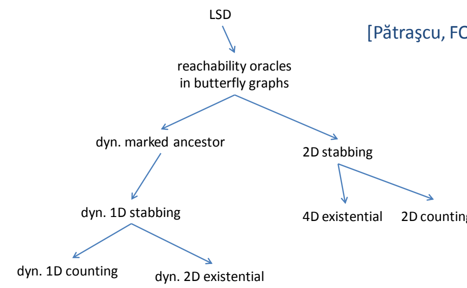
Ω(lg n/lglg n) Lower Bounds

Mihai Pătraşcu
at&t

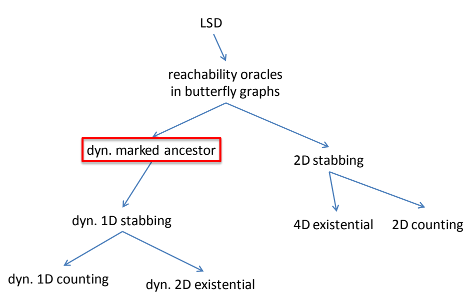
MADALGO Summer School 2010
Tuesday, Morning I

Ω(lg n/lglg n) Bounds

[Pătraşcu, FOCS'08]

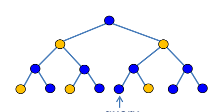


Ω(lg n/lglg n) Bounds



Marked Ancestor

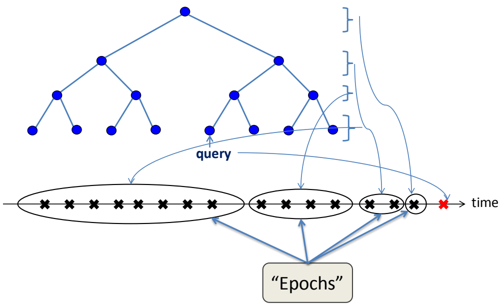
- mark(v) / unmark(v)
- query(v): ∃? marked ancestor



[Alstrup, Husfeldt, Rauhe FOCS'98]
Any data structure with update time $t_u = \lg^{O(1)} n$
requires query time $t_q = \Omega(\lg n / \lg\lg n)$

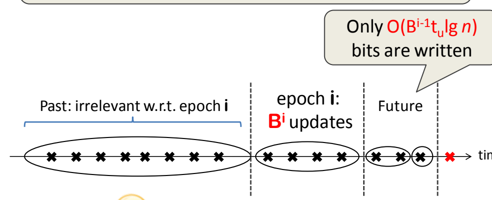
We may assume any branching factor B:
for each used level, ignore (never mark) the next lg B - 1

Hardness Intuition



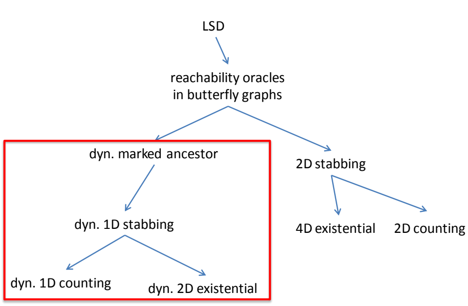
Hardness Intuition

Claim. (∀) i, query needs to read one cell from epoch i
⇒ $t_q = \Omega(\log_B n)$



Let $B \gg t_u \lg n \Rightarrow B^i \gg B^{i-1} t_u \lg n$

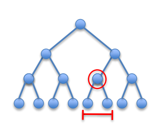
Ω(lg *n*/lglg *n*) Bounds



Reductions

Marked ancestor requires time Ω(lg *n* / lglg *n*)
⇒
Dynamic 1D stabbing requires time Ω(lg *n* / lglg *n*)

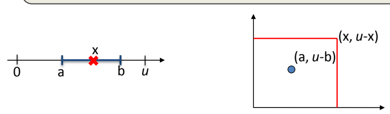
- marked nodes ↦ spanning segment
- query has marked ancestor ⇔ stabs a segment



Reductions

Marked ancestor requires time Ω(lg *n* / lglg *n*)
⇒
Dynamic 1D stabbing requires time Ω(lg *n* / lglg *n*)
⇒
Dynamic 2D dominance existential requires Ω(lg *n* / lglg *n*)

In general: dimensional stabbing in *d* dimensions reduces to dominance queries in 2*d* dimensions

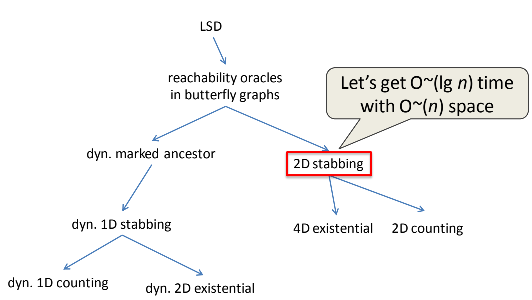


Reductions

Marked ancestor requires time Ω(lg *n* / lglg *n*)
⇒
Dynamic 1D stabbing requires time Ω(lg *n* / lglg *n*)
⇒
Dynamic 2D dominance existential requires Ω(lg *n* / lglg *n*)
Dynamic 1D dominance counting requires Ω(lg *n* / lglg *n*)

Lower bound holds if #marked ancestors ∈ {0,1}
⇒ unweighted version also hard (only parity matters)

Ω(lg *n*/lglg *n*) Bounds

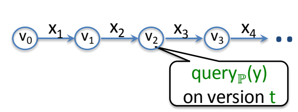


Persistence

Persistent : { dynamic data structures } ↦ { static data structures}
An operator on data structure problems!

Given problem $\mathbb{P}$ ∈ { dynamic d.s.} with update$_{\mathbb{P}}$(x), query$_{\mathbb{P}}$(y)
Persistent($\mathbb{P}$) = the static problem

Preprocess (x_1, x_2, …, x_T) to answer:
- query(y, t): what would query$_{\mathbb{P}}$(y) return after "update$_{\mathbb{P}}$(x_1), …, update$_{\mathbb{P}}$(x_t)"

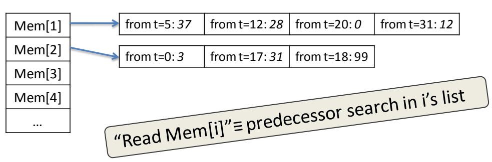


Dynamic ↦ Persistent

Black-box conversion

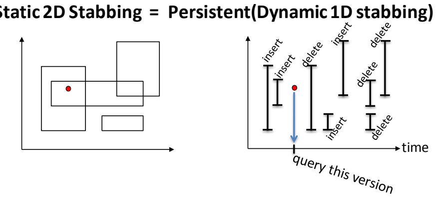
Dynamic data structure: query t_q; update t_u

↦ Persistent data structure: space $O(T \cdot t_u)$; query $O(t_q \cdot \lg\lg T)$



Static 2D Stabbing

Static 2D Stabbing = Persistent(Dynamic 1D stabbing)

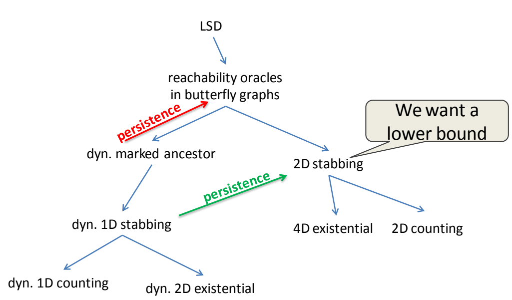


Dynamic 1D stabbing: $O(\lg n)$ by binary search trees

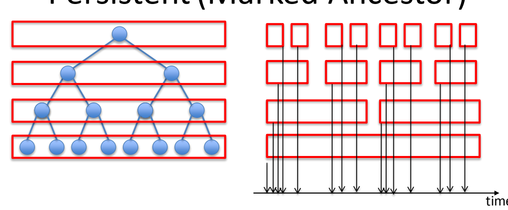
⇒ Static 2D stabbing with space $O(n \lg n)$ and time $O(\lg n \lg\lg n)$

[More care: space $O(n)$, time $O(\lg n / \lg\lg n)$]

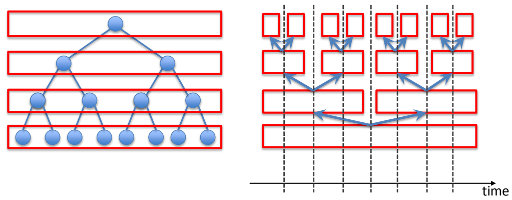
Ω(lg *n*/lglg *n*) Bounds



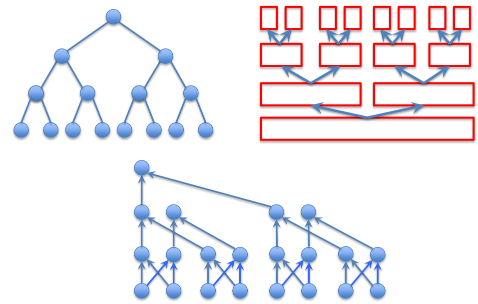
Persistent (Marked Ancestor)



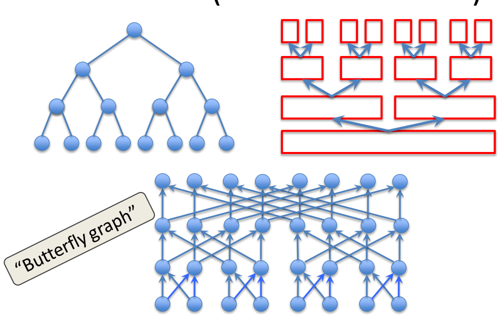
Persistent (Marked Ancestor)



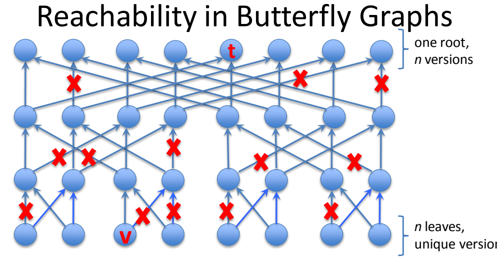
Persistent (Marked Ancestor)



Persistent (Marked Ancestor)

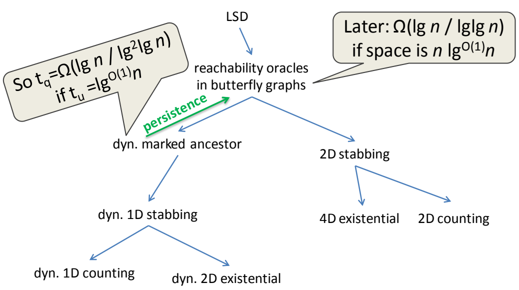


Persistent(Marked Ancestor) = Reachability in Butterfly Graphs

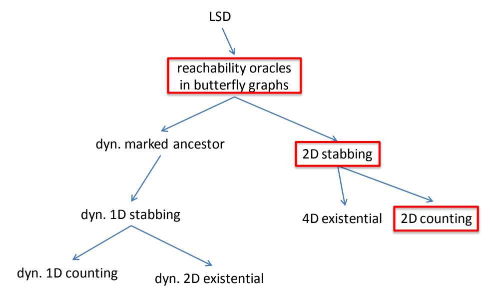


mark(v) ↦ delete edge from v to current parent
query(v,t) ↦ can you reach from v to t?

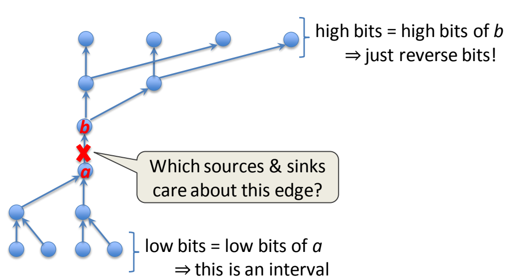
Ω(lg n/lglg n) Bounds



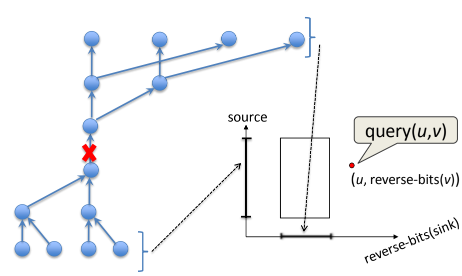
Ω(lg n/lglg n) Bounds



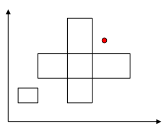
Butterfly Graphs ↦ 2D Stabbing

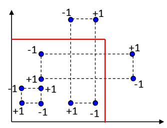


Butterfly Graphs ↦ 2D Stabbing



2D Stabbing ↦ 2D Counting





We can guarantee:

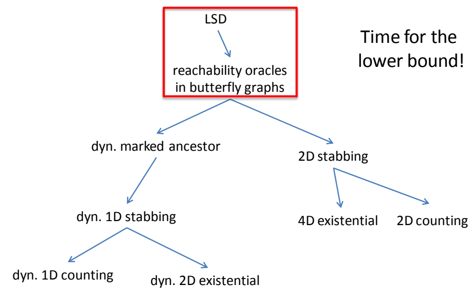
In butterfly query, at most one edge is deleted

⇒ In 2D stabbing, at most one rectangle is stabbed

⇒ Only parity matters in 2D range counting

⇒ unweighted counting also hard

$\Omega(\lg n / \lg\lg n)$ Bounds



Time for the lower bound!

Communication Complexity

Measures: total # bits communicated by Alice, Bob

Indexing

Alice: $i \in \{1, \ldots, n\}$ Output: $v[i]$

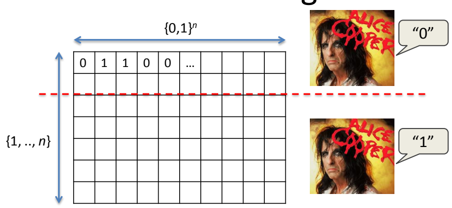
Bob: $v \in \{0,1\}^n$

Trivial solutions:

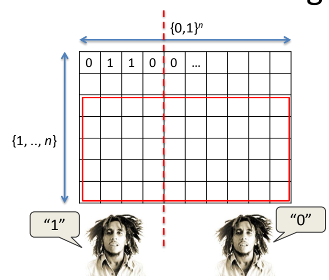
- Alice sends lg n bits; Bob sends 1 bit
- Bob sends n bits; Alice sends 1 bit
- Alice sends a bits; Bob sends $n/2^a$ bits; Alice sends 1 bit

Theorem. If Alice communicate a bits, Bob needs to communicate $\geq n/2^{O(a)}$ bits.

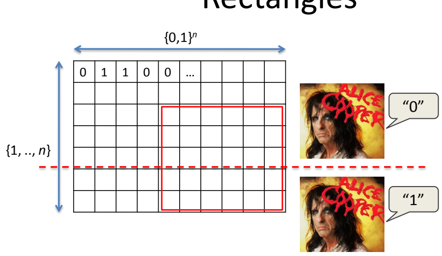
Rectangles



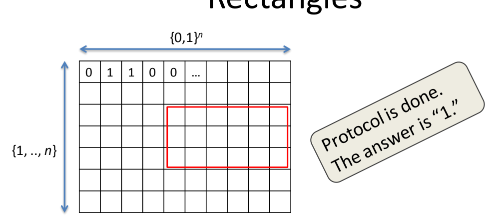
Rectangles



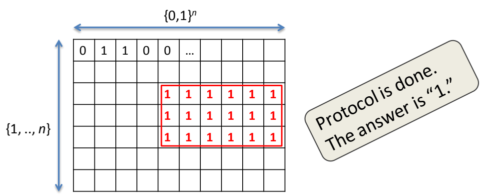
Rectangles



Rectangles



Rectangles



At the end of a protocol, arrive at a *monochromatic* rectangle

Indexing Lower Bound

Theorem. If Alice communicates a bits,
Bob needs to communicate $\geq n/2^{O(a)}$ bits.

If Alice communicates a bits & Bob communicates b bits
$\Rightarrow$ end at monochromatic rectangle $A \times B$
of size $|A| \geq n/2^a$; $|B| \geq 2^n/2^b$

Say output is "1" $\Rightarrow$ $(\forall)$ i $\in A$, v[i] = 1
$\Rightarrow$ at most $2^{n-|A|}$ choices for v
$\Rightarrow 2^n/2^b \leq 2^{n-|A|}$ $\Rightarrow$ $b \geq |A|$ $\Rightarrow$ $b \geq n/2^a$ □

Indexing Lower Bound

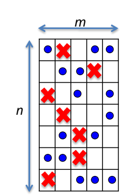
Theorem. If Alice communicates $o(\lg n)$ bits,
Bob needs to communicate $\geq n^{1-o(1)}$ bits.

If Alice communicates a bits & Bob communicates b bits
$\Rightarrow$ end at monochromatic rectangle $A \times B$
of size $|A| \geq n/2^a$; $|B| \geq 2^n/2^b$

Say output is "1" $\Rightarrow$ $(\forall)$ i $\in A$, v[i] = 1
$\Rightarrow$ at most $2^{n-|A|}$ choices for v
$\Rightarrow 2^n/2^b \leq 2^{n-|A|}$ $\Rightarrow$ $b \geq |A|$ $\Rightarrow$ $b \geq n/2^a$ □

<u>L</u>opsided <u>S</u>et <u>D</u>isjointness

- Alice: set S $\subset [n \cdot m]$, $|S| = n$
- Bob: set T $\subseteq [n \cdot m]$, $|T| \leq n \cdot m$
- Output: S $\cap$ T = $\emptyset$?



Special case: LSD = $n \times$ Indexing(m)
A = { ✖ ✖ ✖ } B = { ● ● ● }

Theorem. If Alice communicates $o(n \lg m)$ bits,
Bob needs to communicate $\geq n \cdot m^{1-o(1)}$ bits

Lopsided Set Disjointness

Inputs: Alice $(i_1, \ldots, i_n) \in [m]^n$, Bob $(v_1, \ldots, v_n) \in (\{0,1\}^m)^n$

Alice sends $n \cdot a$ bits, Bob sends $n \cdot b$ bits

$\Rightarrow$ rectangle of 1's $A \times B$ of size:

$(\forall)k, v_k[i_k] = 0$

$|A| \geq m^n / 2^{na} = (m / 2^a)^n$

$|B| \geq 2^{mn} / 2^{nb} = (2^m / 2^b)^n$

$A \subseteq \pi_1(A) \times \cdots \pi_n(A) \Rightarrow$ for ½n+1 coordinates $|\pi_j(A)| \geq m / 2^{2a}$

$B \subseteq \pi_1(B) \times \cdots \pi_n(B) \Rightarrow$ for ½n+1 coordinates $|\pi_j(B)| \geq 2^m / 2^{2b}$

So $(\exists)$j: $|\pi_j(A)| \geq m / 2^{2a}$ & $|\pi_j(B)| \geq 2^m / 2^{2b}$

$\pi_j(A) \times \pi_j(B)$ monochromatic rectangle for Indexing.

Lopsided Set Disjointness

Indexing: If Alice communicates o(lg n) bits,
Bob needs to communicate $\geq n^{1-o(1)}$ bits.

$\Rightarrow$

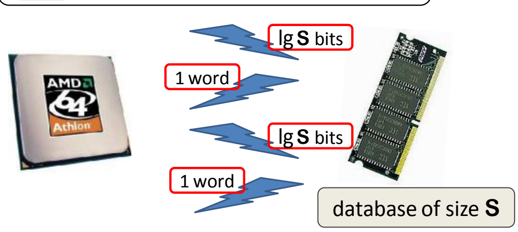
LSD: If Alice communicates o(n lg m) bits,
Bob needs to communicate $\geq n \cdot m^{1-o(1)}$ bits.

So $(\exists)$j: $|\pi_j(A)| \geq m / 2^{2a}$ & $|\pi_j(B)| \geq 2^m / 2^{2b}$

$\pi_j(A) \times \pi_j(B)$ monochromatic rectangle for Indexing.

Communication Complexity → Data Structures

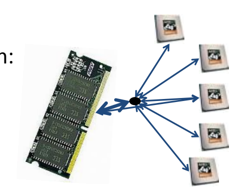
Then what's the difference between
S=O(**n**) and **S**=O(**n**^{2}) ?



Communication Complexity → Data Structures

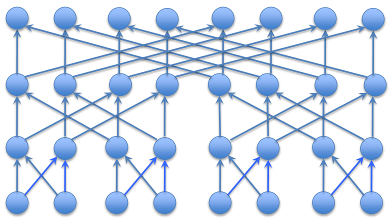
Processor → memory bandwidth:

- one processor: lg **S**
- **k** processors: lg $\binom{S}{k}$ ≈ **k** lg $\frac{S}{k}$
  amortized lg(**S**/**k**) / processor

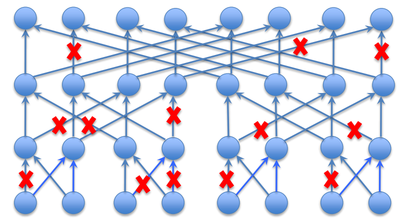


	S=O(n)	S=O(n^2)
k = 1	lg n	2lg n
k = n/lg n	lglg n	≈ lg n

LSD ➞ Reachability Oracles

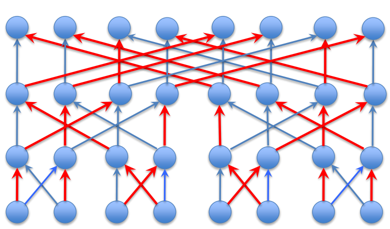


LSD ➞ Reachability Oracles



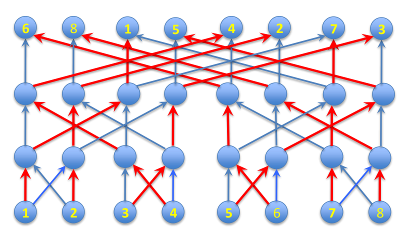
T = {✗} = deleted edges

LSD ➞ Reachability Oracles



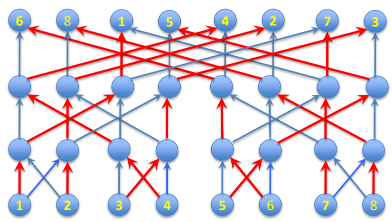
T = {✗} = deleted edges
S = {↗} = one out-edge / node

LSD ➞ Reachability Oracles



T = {✗} = deleted edges
S = {↗} = one out-edge / node

LSD → Reachability Oracles



n queries: source *k* → sink *k*

S ∩ T ≠ ∅ ⇒ some query answer false

LSD → Reachability Oracles

Butterfly ⇔ Set disjointness

- width *n*
- degree $B \geq \lg n$

Input size $N = n \cdot B$

Space $S = N \lg^{O(1)} N = n B \lg^{O(1)} n$

Alice sends $t \cdot n \lg(S/n)$
$= n \cdot t \cdot O(\lg B)$

Bob sends $n \cdot t \cdot \lg n$

Set disjointness

- $|S| = n \log_B n$
- $|T| = n \cdot B = |S| \cdot (B/\log_B n)$

Thus: $m = B/\log_B n \approx B/\lg B$

If Alice sends $o(|S| \lg m)$
$= o(n \log_B n \lg B)$

Bob must send $|S| \cdot m^{1-o(1)}$
$= n \cdot B^{1-o(1)}$

LSD → Reachability Oracles

If $t = o(\log_B n)$ ⇒ $t \lg n \geq B^{1-o(1)}$
Set $B = \lg^3 n$ ⇒ contradiction.
Thus $t = \Omega(\log n / \lg\lg n)$

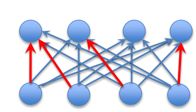
Alice sends $t \cdot n \lg(S/n)$
$= n \cdot t \cdot O(\lg B)$

Bob sends $n \cdot t \cdot \lg n$

If Alice sends $o(|S| \lg m)$
$= o(n \log_B n \lg B)$

Bob must send $|S| \cdot m^{1-o(1)}$
$= n \cdot B^{1-o(1)}$

How to Get a Matching



Universe B^2; $|S| = B$; $|T| \leq B^2$

S edges form a matching ⇔ perfect hashing $[B^2] \mapsto [B]$

O(B) bits to describe the perfect hash function
⇒ negligible communication