Parallel Comparison Tree Model

A parallel comparison tree (with parallelism p) is a 3^p-ary tree.

- each internal node represents a set of p comparisons btw. p pairs (not necessarily distinct)
- a leaf v corresponds to a unique permutation that is valid for all the comparisons on the path from the root to v
- the number of parallel steps is the height of the tree

Parallel Comparison Tree Model

A parallel comparison tree (with parallelism p) is a 3^p-ary tree.

- each internal node represents a set of p comparisons btw. p pairs (not necessarily distinct)
- a leaf v corresponds to a unique permutation that is valid for all the comparisons on the path from the root to v
- the number of parallel steps is the height of the tree

Parallel Comparison Tree Model

A parallel comparison tree (with parallelism p) is a 3^p-ary tree.

- each internal node represents a set of p comparisons btw. p pairs (not necessarily distinct)
- a leaf v corresponds to a unique permutation that is valid for all the comparisons on the path from the root to v
- the number of parallel steps is the height of the tree

Comparison PRAM

A comparison PRAM is a PRAM where we can only compare the input elements;

- we cannot view them as strings
- we cannot do calculations on them

A lower bound for the comparison tree with parallelism p directly carries over to the comparison PRAM with p processors.

Comparison PRAM

A comparison PRAM is a PRAM where we can only compare the input elements;

- we cannot view them as strings
- we cannot do calculations on them

A lower bound for the comparison tree with parallelism p directly carries over to the comparison PRAM with p processors.

Comparison PRAM

A comparison PRAM is a PRAM where we can only compare the input elements;

- we cannot view them as strings
- we cannot do calculations on them

A lower bound for the comparison tree with parallelism p directly carries over to the comparison PRAM with p processors.

Comparison PRAM

A comparison PRAM is a PRAM where we can only compare the input elements;

- we cannot view them as strings
- we cannot do calculations on them

A lower bound for the comparison tree with parallelism p directly carries over to the comparison PRAM with p processors.

A Lower Bound for Searching

Theorem 1
Given a sorted table X of n elements and an element y. Searching for y in X requires $\Omega(\frac{\log n}{\log(p+1)})$ steps in the parallel comparsion tree with parallelism $p < n$.

A Lower Bound for Maximum

Theorem 2
A graph G with m edges and n vertices has an independent set on at least $\frac{n^2}{2m+n}$ vertices.

base case ($n = 1$)

- The only graph with one vertex has $m = 0$, and an independent set of size 1.

A Lower Bound for Maximum

Theorem 2

A graph G with m edges and n vertices has an independent set on at least $\frac{n^2}{2m+n}$ vertices.

base case ($n = 1$)

- The only graph with one vertex has $m = 0$, and an independent set of size 1.

A Lower Bound for Maximum

Theorem 2
A graph G with m edges and n vertices has an independent set on at least $\frac{n^2}{2m+n}$ vertices.

base case ($n = 1$)

- The only graph with one vertex has $m = 0$, and an independent set of size 1.

induction step $(1, \dots, n \to n+1)$

- Let G be a graph with $n+1$ vertices, and v a node with minimum degree (d).
- Let G' be the graph after deleting v and its adjacent vertices in G.
- $n' = n - (d+1)$
- $m' \le m - \frac{d}{2}(d+1)$ as we remove $d+1$ vertices, each with degree at least d
- In G' there is an independent set of size $((n')^2/(2m'+n'))$.
- By adding v we obtain an indepent set of size

$$1 + \frac{(n')^2}{2m'+n'} \ge \frac{n^2}{2m+n}$$

induction step ($1, \dots, n \to n+1$)

- Let G be a graph with $n+1$ vertices, and v a node with minimum degree (d).
- Let G' be the graph after deleting v and its adjacent vertices in G.
- $n' = n - (d+1)$
- $m' \leq m - \frac{d}{2}(d+1)$ as we remove $d+1$ vertices, each with degree at least d
- In G' there is an independent set of size $((n')^2/(2m'+n'))$.
- By adding v we obtain an indepent set of size

$$1 + \frac{(n')^2}{2m'+n'} \geq \frac{n^2}{2m+n}$$

induction step ($1, \ldots, n \rightarrow n+1$)

- Let G be a graph with $n+1$ vertices, and v a node with minimum degree (d).
- Let G' be the graph after deleting v and its adjacent vertices in G.
- $n' = n - (d+1)$
- $m' \leq m - \frac{d}{2}(d+1)$ as we remove $d+1$ vertices, each with degree at least d
- In G' there is an independent set of size $((n')^2/(2m'+n'))$.
- By adding v we obtain an indepent set of size

$$1 + \frac{(n')^2}{2m'+n'} \geq \frac{n^2}{2m+n}$$

induction step ($1, \ldots, n \to n+1$)

- Let G be a graph with $n+1$ vertices, and v a node with minimum degree (d).
- Let G' be the graph after deleting v and its adjacent vertices in G.
- $n' = n - (d+1)$
- $m' \leq m - \frac{d}{2}(d+1)$ as we remove $d+1$ vertices, each with degree at least d
- In G' there is an independent set of size $((n')^2/(2m'+n'))$.
- By adding v we obtain an indepent set of size

$$1 + \frac{(n')^2}{2m'+n'} \geq \frac{n^2}{2m+n}$$

induction step $(1, \ldots, n \to n + 1)$

- Let G be a graph with $n + 1$ vertices, and v a node with minimum degree (d).
- Let G' be the graph after deleting v and its adjacent vertices in G.
- $n' = n - (d + 1)$
- $m' \leq m - \frac{d}{2}(d + 1)$ as we remove $d + 1$ vertices, each with degree at least d
- In G' there is an independent set of size $((n')^2/(2m' + n'))$.
- By adding v we obtain an indepent set of size

$$1 + \frac{(n')^2}{2m' + n'} \geq \frac{n^2}{2m + n}$$

induction step ($1, \dots, n \to n+1$)

- Let G be a graph with $n+1$ vertices, and v a node with minimum degree (d).
- Let G' be the graph after deleting v and its adjacent vertices in G.
- $n' = n - (d+1)$
- $m' \leq m - \frac{d}{2}(d+1)$ as we remove $d+1$ vertices, each with degree at least d
- In G' there is an independent set of size $((n')^2/(2m'+n'))$.
- By adding v we obtain an indepent set of size

$$1 + \frac{(n')^2}{2m'+n'} \geq \frac{n^2}{2m+n}$$

A Lower Bound for Maximum

Theorem 3
Computing the maximum of n elements in the comparison tree requires $\Omega(\log\log n)$ steps whenever the degree of parallelism is $p \le n$.

Theorem 4
Computing the maximum of n elements requires $\Omega(\log\log n)$ steps on the comparison PRAM with n processors.

A Lower Bound for Maximum

Theorem 3
Computing the maximum of n elements in the comparison tree requires $\Omega(\log\log n)$ steps whenever the degree of parallelism is $p \leq n$.

Theorem 4
Computing the maximum of n elements requires $\Omega(\log\log n)$ steps on the comparison PRAM with n processors.

An adversary can specify the input such that at the end of the $(i+1)$-st step the maximum lies in a set C_{i+1} of size s_{i+1} such that

- no two elements of C_{i+1} have been compared
- $s_{i+1} \geq \frac{s_i^2}{2p+c_i}$

An adversary can specify the input such that at the end of the $(i+1)$-st step the maximum lies in a set C_{i+1} of size s_{i+1} such that

- no two elements of C_{i+1} have been compared
- $s_{i+1} \geq \frac{s_i^2}{2p+c_i}$

Theorem 5
The selection problem requires $\Omega(\log n / \log\log n)$ steps on a comparison PRAM.

not proven yet

A Lower Bound for Merging

The (k, s)-merging problem, asks to merge k pairs of subsequences $A^1, \dots, A^k$ and $B^1, \dots, B^k$ where we know that all elements in $A^i \cup B^i$ are smaller than elements in $A^j \cup B^j$ for $(i < j)$.

A Lower Bound for Merging

Lemma 6

Suppose we are given a parallel comparison tree with parallelism p *to solve the* (k,s) *merging problem. After the first step an adversary can specify the input such that an arbitrary* (k',s') *merging problem has to be solved, where*

$$k' = \frac{3}{4}\sqrt{pk}$$

$$s' = \frac{s}{4}\sqrt{\frac{k}{p}}$$

A Lower Bound for Merging

Partition $A^i s$ and $B^i s$ into blocks of length roughly s/ℓ; hence ℓ blocks.

Define an $\ell \times \ell$ binary matrix M^i, where M^i_{xy} is 0 iff the parallel step **did not** compare an element from A^i_x with an element from B^i_y.

The matrix has $2\ell - 1$ diagonals.

A Lower Bound for Merging

Partition A^is and B^is into blocks of length roughly s/ℓ; hence ℓ blocks.

Define an $\ell \times \ell$ binary matrix M^i, where M^i_{xy} is 0 iff the parallel step **did not** compare an element from A^i_x with an element from B^i_y.

The matrix has $2\ell - 1$ diagonals.

A Lower Bound for Merging

Partition $A^i s$ and $B^i s$ into blocks of length roughly s/ℓ; hence ℓ blocks.

Define an $\ell \times \ell$ binary matrix M^i, where M^i_{xy} is 0 iff the parallel step **did not** compare an element from A^i_x with an element from B^i_y.

The matrix has $2\ell - 1$ diagonals.

Choose for every i the diagonal of M^i that has most zeros.

Pair all $A^i_{j+d_i}, B^i_j$, (where $d_i \in \{-(\ell-1), \dots, \ell-1\}$ specifies the chosen diagonal) for which the entry in M^i is zero.

We can choose value s.t. elements for the j-th pair along the diagonal are **all** smaller than for the $(j+1)$-th pair.

Hence, we get a (k', s') problem.

Choose for every i the diagonal of M^i that has most zeros.

Pair all $A^i_{j+d_i}, B^i_j$, (where $d_i \in \{-(\ell-1), \dots, \ell-1\}$ specifies the chosen diagonal) for which the entry in M^i is zero.

We can choose value s.t. elements for the j-th pair along the diagonal are **all** smaller than for the $(j+1)$-th pair.

Hence, we get a (k', s') problem.

Choose for every i the diagonal of M^i that has most zeros.

Pair all $A^i_{j+d_i}, B^i_j$, (where $d_i \in \{-(\ell-1), \dots, \ell-1\}$ specifies the chosen diagonal) for which the entry in M^i is zero.

We can choose value s.t. elements for the j-th pair along the diagonal are **all** smaller than for the $(j+1)$-th pair.

Hence, we get a (k', s') problem.

Choose for every i the diagonal of M^i that has most zeros.

Pair all $A^i_{j+d_i}, B^i_j$, (where $d_i \in \{-(\ell-1), \ldots, \ell-1\}$ specifies the chosen diagonal) for which the entry in M^i is zero.

We can choose value s.t. elements for the j-th pair along the diagonal are **all** smaller than for the $(j+1)$-th pair.

Hence, we get a (k', s') problem.

Choose for every i the diagonal of M^i that has most zeros.

Pair all $A^i_{j+d_i}, B^i_j$, (where $d_i \in \{-(\ell-1), \ldots, \ell-1\}$ specifies the chosen diagonal) for which the entry in M^i is zero.

We can choose value s.t. elements for the j-th pair along the diagonal are **all** smaller than for the $(j+1)$-th pair.

Hence, we get a (k', s') problem.

How many pairs do we have?

- there are $k\ell$ blocks in total
- there are $k \cdot \ell^2$ matrix entries in total
- there are at least $k \cdot \ell^2 - p$ zeros.
- choosing a random diagonal (same for every matrix M^i) hits at least

$$\frac{k\ell^2 - p}{2\ell - 1} \geq \frac{k\ell}{2} - \frac{p}{2\ell}$$

zeroes.
- Choosing $\ell = 2\sqrt{\frac{p}{k}}$ gives

$$k' \geq \frac{3}{4}\sqrt{pk} \text{ and } s' = \lfloor \frac{s}{\ell} \rfloor \geq \frac{s}{2\ell} = \frac{s}{4}\sqrt{\frac{k}{p}}$$

where we assume $\frac{s}{\ell} \geq 2$.

How many pairs do we have?

- there are $k\ell$ blocks in total
- there are $k \cdot \ell^2$ matrix entries in total
- there are at least $k \cdot \ell^2 - p$ zeros.
- choosing a random diagonal (same for every matrix M^i) hits at least
$$\frac{k\ell^2 - p}{2\ell - 1} \geq \frac{k\ell}{2} - \frac{p}{2\ell}$$
zeroes.
- Choosing $\ell = 2\sqrt{\frac{p}{k}}$ gives
$$k' \geq \frac{3}{4}\sqrt{pk} \text{ and } s' = \lfloor \frac{s}{\ell} \rfloor \geq \frac{s}{2\ell} = \frac{s}{4}\sqrt{\frac{k}{p}}$$
where we assume $\frac{s}{\ell} \geq 2$.

How many pairs do we have?

- there are $k\ell$ blocks in total
- there are $k \cdot \ell^2$ matrix entries in total
- there are at least $k \cdot \ell^2 - p$ zeros.
- choosing a random diagonal (same for every matrix M^i) hits at least

$$\frac{k\ell^2 - p}{2\ell - 1} \geq \frac{k\ell}{2} - \frac{p}{2\ell}$$

zeroes.

- Choosing $\ell = 2\sqrt{\frac{p}{k}}$ gives

$$k' \geq \frac{3}{4}\sqrt{pk} \text{ and } s' = \lfloor \frac{s}{\ell} \rfloor \geq \frac{s}{2\ell} = \frac{s}{4}\sqrt{\frac{k}{p}}$$

where we assume $\frac{s}{\ell} \geq 2$.

How many pairs do we have?

- there are $k\ell$ blocks in total
- there are $k \cdot \ell^2$ matrix entries in total
- there are at least $k \cdot \ell^2 - p$ zeros.
- choosing a random diagonal (same for every matrix M^i) hits at least

$$\frac{k\ell^2 - p}{2\ell - 1} \geq \frac{k\ell}{2} - \frac{p}{2\ell}$$

zeroes.

- Choosing $\ell = 2\sqrt{\frac{p}{k}}$ gives

$$k' \geq \frac{3}{4}\sqrt{pk} \text{ and } s' = \lfloor \frac{s}{\ell} \rfloor \geq \frac{s}{2\ell} = \frac{s}{4}\sqrt{\frac{k}{p}}$$

where we assume $\frac{s}{\ell} \geq 2$.

How many pairs do we have?

- there are $k\ell$ blocks in total
- there are $k \cdot \ell^2$ matrix entries in total
- there are at least $k \cdot \ell^2 - p$ zeros.
- choosing a random diagonal (same for every matrix M^i) hits at least
$$\frac{k\ell^2 - p}{2\ell - 1} \geq \frac{k\ell}{2} - \frac{p}{2\ell}$$
zeroes.
- Choosing $\ell = 2\sqrt{\frac{p}{k}}$ gives
$$k' \geq \frac{3}{4}\sqrt{pk} \text{ and } s' = \lfloor \frac{s}{\ell} \rfloor \geq \frac{s}{2\ell} = \frac{s}{4}\sqrt{\frac{k}{p}}$$
where we assume $\frac{s}{\ell} \geq 2$.

Lemma 7

Let $T(k, s, p)$ be the number of parallel steps required on a comparison tree to solve the (k, s) merging problem. Then

$$T(k, p, s) \geq \frac{1}{4} \log \frac{\log \frac{p}{k}}{\log \frac{p}{ks}}$$

provided that $p \geq 2ks$ and $p \leq ks^2/4$

Induction Step:

Assume that

$$T(k', s', p) \geq \frac{1}{4} \log \frac{\log \frac{p}{k'}}{\log \frac{p}{k's'}}$$

Induction Step:

Assume that

$$T(k', s', p) \geq \frac{1}{4} \log \frac{\log \frac{p}{k'}}{\log \frac{p}{k's'}}$$

$$\geq \frac{1}{4} \log \frac{\log \frac{4}{3}\sqrt{\frac{p}{k}}}{\log \frac{16}{3}\frac{p}{ks}}$$

Induction Step:

Assume that

$$\begin{aligned} T(k',s',p) &\geq \frac{1}{4}\log\frac{\log\frac{p}{k'}}{\log\frac{p}{k's'}} \\ &\geq \frac{1}{4}\log\frac{\log\frac{4}{3}\sqrt{\frac{p}{k}}}{\log\frac{16}{3}\frac{p}{ks}} \\ &\geq \frac{1}{4}\log\frac{\frac{1}{2}\log\frac{p}{k}}{7\log\frac{p}{ks}} \end{aligned}$$

Induction Step:

Assume that

$$
\begin{aligned}
T(k',s',p) &\geq \frac{1}{4}\log\frac{\log\frac{p}{k'}}{\log\frac{p}{k's'}}\\
&\geq \frac{1}{4}\log\frac{\log\frac{4}{3}\sqrt{\frac{p}{k}}}{\log\frac{16}{3}\frac{p}{ks}}\\
&\geq \frac{1}{4}\log\frac{\frac{1}{2}\log\frac{p}{k}}{7\log\frac{p}{ks}}\\
&\geq \frac{1}{4}\log\frac{\log\frac{p}{k}}{\log\frac{p}{ks}}-1
\end{aligned}
$$

Induction Step:

Assume that

$$\begin{aligned} T(k',s',p) &\geq \frac{1}{4}\log\frac{\log\frac{p}{k'}}{\log\frac{p}{k's'}} \\ &\geq \frac{1}{4}\log\frac{\log\frac{4}{3}\sqrt{\frac{p}{k}}}{\log\frac{16}{3}\frac{p}{ks}} \\ &\geq \frac{1}{4}\log\frac{\frac{1}{2}\log\frac{p}{k}}{7\log\frac{p}{ks}} \\ &\geq \frac{1}{4}\log\frac{\log\frac{p}{k}}{\log\frac{p}{ks}} - 1 \end{aligned}$$

This gives the induction step.

Theorem 8

Merging requires at least $\Omega(\log\log n)$ time on a CRCW PRAM with n processors.