List Ranking

Pointer Jumping (Recursive doubling)

$O(\log n)$ time n processors

EREW PRAM

$O(n)$ cost algorithm

$O(\log n)$ time using $\frac{n}{\log n}$ processors

EREW PRAM

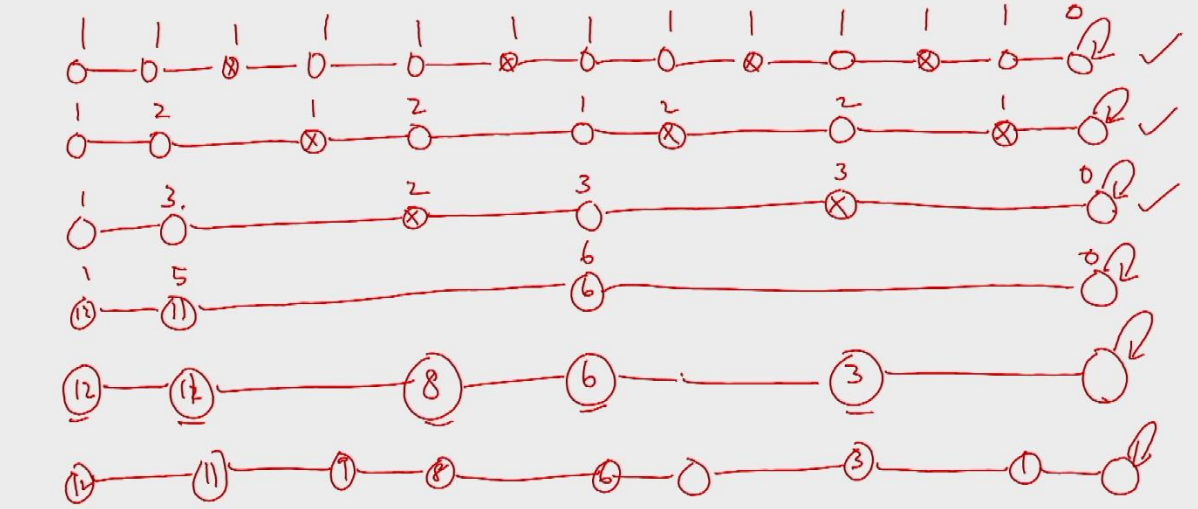
$O(n)$ optimal

Vertex reduction

Linked list of length n
in Array of size n

$n \rightarrow n/\log n$ no. vertices in the list

$O(\log n)$ time



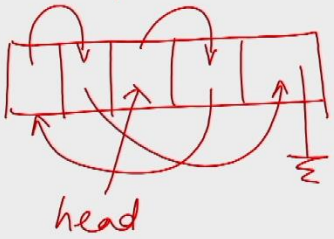
$O\left(\log \frac{n}{\log n}\right) = O(\log n)$ time $\frac{n}{\log n}$ procs

Phase 1:
- $O(\log n)$ several iterations
- $O(1)$ time iterations: independent set removed from the list
- list length $\leq n/\log n$

Phase 2: Pointer Jumping
- several iterations

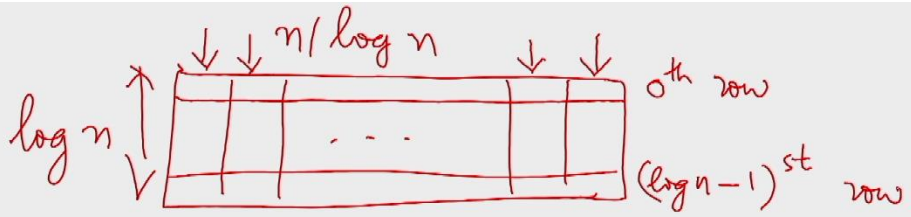
Vertex Reduction

Physical representation



1D-array of length n

Visualize the array as a 2-D array



- inactive
- active
- ruler
- subject
- removed

} 5 labels for the vertices

iteration for the i^{th} node

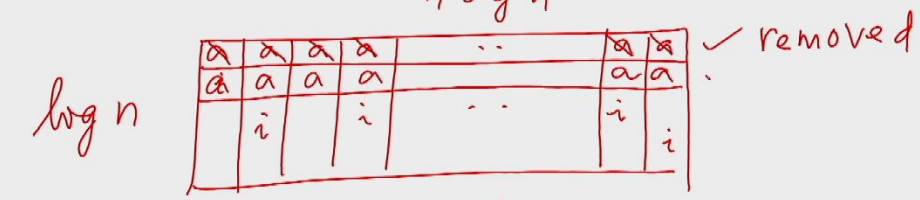
if (the node is labelled ruler)
    splice out the next subject
if (no more subjects are left)
    turns active

if (active and isolated)
    splice it self out & advance
} lucky

if (active and non isolated)
    participate in subject
        mker election

---

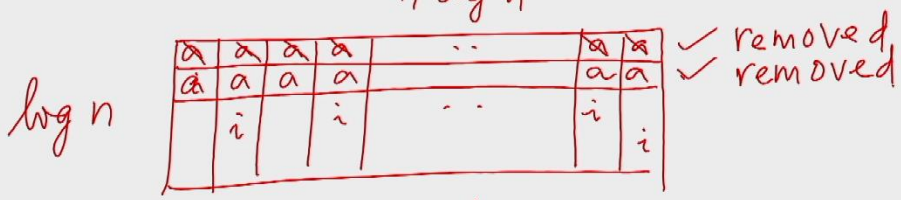
lucky all the time



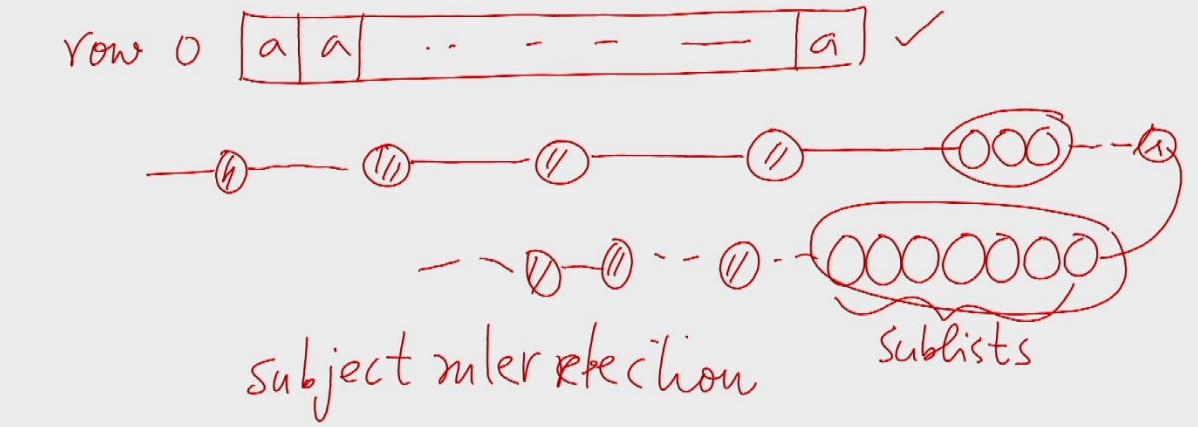
isolated active node
is one that is not adjacent to
active nodes

lucky again:
we get an IS again

lucky all the time



isolated active node
is one that is not adjacent to
active nodes



if (active and isolated)
splice it self out & advance
(removed)
} lucky

if (active and non isolated)
participate in subject
mler election
& move forward if its node has
been elected a subject

(≥2) ~~2nd~~ phase

r (S) s s s r [S]

if (ruler)

→ then remove the 1st subject

if no further subject left

turn active

if (active)

if isolated sphere out ↓

else SRE, ↓ if chosen subjects

$O(\log n)$ phases

before # vertices $\leq n/\log n$

Subject-ruler election

1. colour the sublist using loglog n colours
Pick every local minimum
Break the list before the local minima