1. $\forall v \in V$. find the smallest nbr

2.1. if this is smaller than v, $p(v)$

2.2. check if v has got children

2.3 if v has no child, no parent

v picks some nbr as the parent

3. Edge plugging

4. Pointer Jumping

$$\left. p(v) = p(p(v)) \right\} O(\log n) \text{ steps}$$

every tree is a <u>star</u>

supervertex

5. Renaming.

if r is the root of v's tree
v merges with r

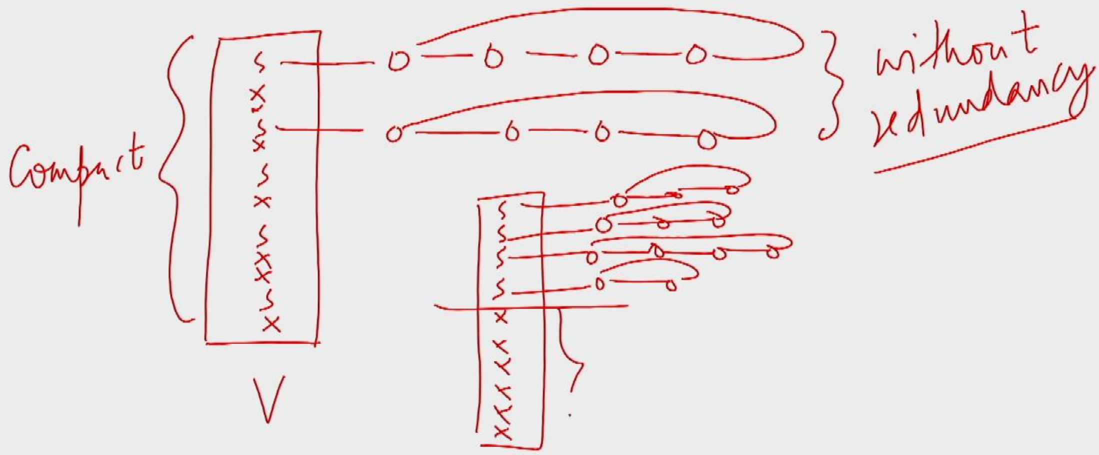
6. Redundancy Removal

6.1 ~~Son~~ List Rank all adj list
copied into an array

6.2 Cole's merge sort
on each adjacency list (array)

$$\cdots\ \underbrace{\overset{\checkmark}{[4,3]}\,\overset{\times}{[4,3]}}_{1}\ \ \underbrace{\overset{\times}{[4,4]}\,\overset{\times}{[4,4]}\,\overset{\times}{[4,4]}}_{0}\ \ \underbrace{\overset{\checkmark}{[4,5]}\,\overset{\times}{[4,5]}\,\overset{\times}{[4,5]}}_{1}\ \cdots$$

Compaction using prefix sums
$O(\log n)$ time.



We are ready for the next iteration

$O(\log n)$ time $n+m$ processors

Look at the hooking step

- every vertex gets either a parent / a child

$n/2$ supervertices

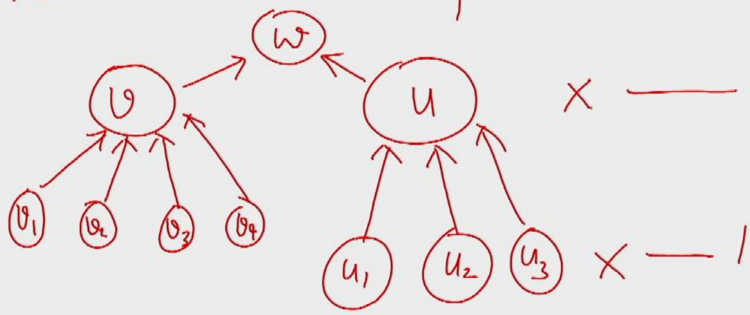
\# vertices $\downarrow$ by a factor 2

$\Longrightarrow$

$O(\log n)$ iterations

$\Rightarrow O(\log^2 n)$ time to reduce the graph to one with no edges

We need a star for each CC



One tree for each CC

The root of this tree
is the vertex in that CC
that survived the last
item

Pointer Jumping

$O(\log n)$ steps.

every cc has a star graph.

$O(\log^2 n)$ time $n+m$ prcrs

CREW PRAM

Vertex Colouring of graphs

Minimum # of colours

: NP-complete

Δ-degree graph

each vertex has $\leq \Delta$ neighbours

$(\Delta+1)$-colouring possible

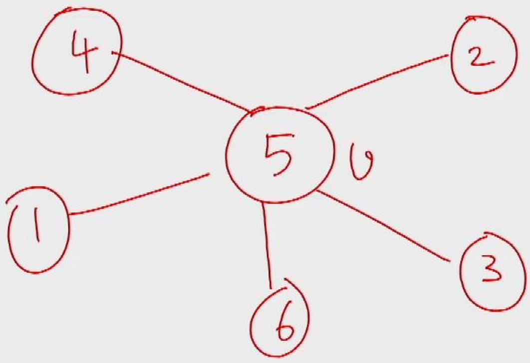
Colours : $1, 2, \ldots, \Delta+1$

n vertex graph.

max. vertex degree is Δ.

remove a vertex. (v)

Recursively $(\Delta+1)$-colour $G-v$



Δ nbrs

Colours : $1, 2, \ldots, \Delta+1$

n vertex graph.

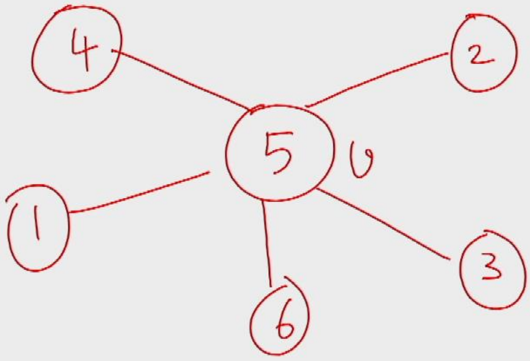
max. vertex degree is Δ.

remove a vertex. (v)

Recursively $(\Delta+1)$-colour $G-v$

Put v back in

colour v



Δ nbrs

$\underline{O(n\Delta)}$

$O(n)$ if $\Delta = O(1)$

$(\Delta+1)$ – vertex colour
a Δ-degree graph
on EREW PRAM

$\Delta = O(1)$

Step 1

$\boxed{v} - o \to o \to o \to o \to o$

for $v \in V$ pardo

for $1 \leq i \leq \Delta$

(1.1) if $p(v) = ?$, propose to the i^{th} nbr of v

(1.2) if $s(v) = ?$ choose one from the proposals rcvd, if any

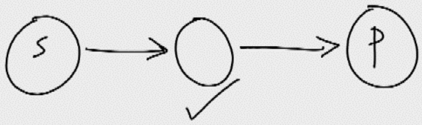
(1.3) if v's proposal to w_i has been accepted $p(v) = w_i$

A vertex has ≤ 1 successor & ≤ 1 predecessor.

if all of v's proposals are turned down & all of v's nbrs get accepted elsewhere

v has no successor & no predecessor

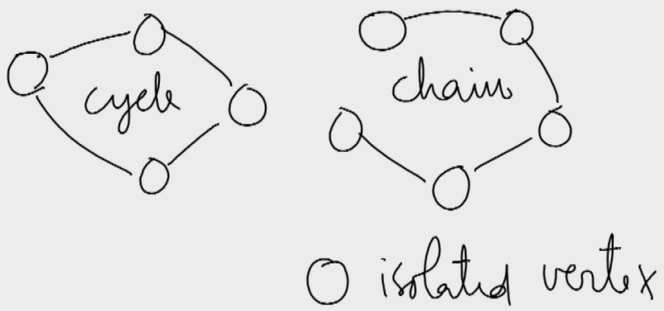
Define L_p as the subgraph induced by the successor-predecessor ptrs.



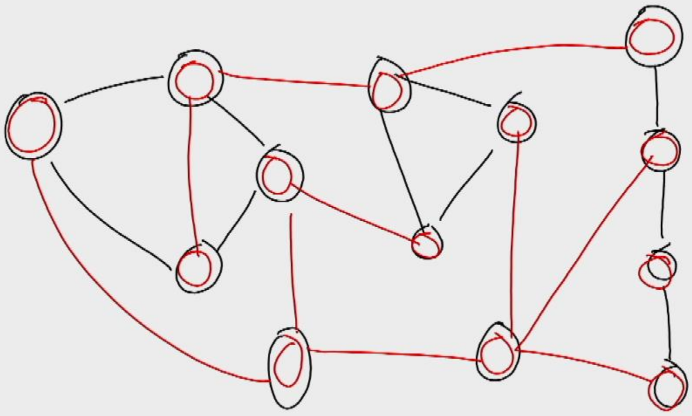
degree ≤ 2

degree 2

L_p is a degree-2 graph.



2. Form $G - L_p$



$G - L_p$ & L_p have the same vertex sets

but disjoint edge sets

3. Colour L_p with 3-colours

List colouring algorithm

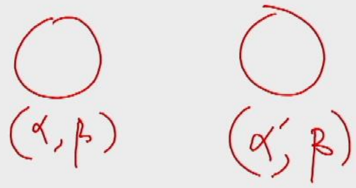
4. Recursively $(\Delta+1)$-colour $G - L_p$

say vertex v

gets α in the $(\Delta+1)$-colouring

β in the 3-colouring

(α, β) : a colour $\underline{\underline{G}}$

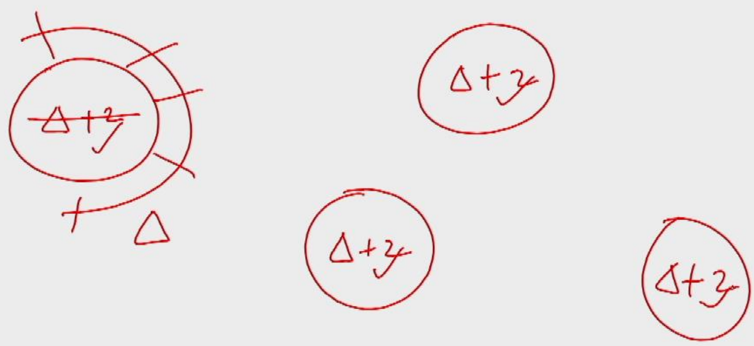


This colouring of G

uses $3(\Delta+1)$ colours

Consider $\Delta+2$ to $3(\Delta+1)$

one by one



$\Delta+3$ $\Delta+3$ $\Delta+3$

$\Delta+3$

$\Delta+2$

$3(\Delta+1)$

$O(\Delta)$

in $G - L_p$,

~~an~~ i

a vertex that is isolated in L_p

has a degree of $\underline{\underline{\Delta}}$

in $(\Delta-1)$-levels of recursion,
this vertex is the only
nbr that its nbrs have

$O(\log^{(k)} n)$ time
with $n/\log^{(k)} n$ processor
if $\Delta = O(1)$