Sorting

Compartor Networks

(EREW PRAM)

$O(\log^2 n)$ time n processors

Batcher's

$\Theta(n \log n)$

Cost $O(n \log^2 n)$ not optimal

Divide Conquer

Merge Sorts

A New Merge Sort

Two sorted arrays of size n

$n/\log\log n$ processor

Divide into 2 segments

Sort each recursively

Merge them

$$T(n) = T\left(\frac{n}{2}\right) + \log\log n$$

$$= T\left(\frac{n}{4}\right) + \log\log \frac{n}{2} + \log\log n$$

$$= \vdots$$

$$= T\left(\frac{n}{2^k}\right) + \log\log \frac{n}{2^{k-1}} + \cdots + \log\log n$$

$$= O(\log n \cdot \log\log n)$$

$n/\log\log n$ processors

$O(\log n \cdot \log\log n)$ time

optimal.

$$\left.\begin{array}{l} O(\log^2 n) \\ O(\log n \cdot \log\log n) \end{array}\right\} \text{time}$$

Cole's Merge Sort

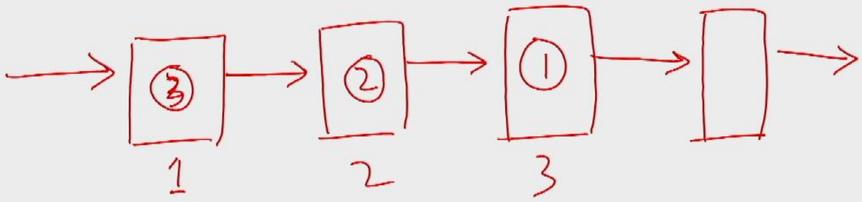
$O(\log n)$ time using

n processors

CREW PRAM

(EREW)

Pipelined Algorithm



$\log n$ Root
:
3
2
1
0 leaves

An array of n elements

$n = 2^k$ (else pad)

Binary tree on top of this array

n leaves

$\log n$ levels above the leaves

At each node of the binary tree, we have 2 array

$$\left.\begin{array}{l}\text{Cache}\\\text{Sample}\end{array}\right\} 2^{\ell}$$ where ℓ is the level of the node

Leaves ~~Roots~~ have

$|\text{Cache}| = |\text{sample}| = 1$

Parents of leaves — 2

nodes of level ℓ — 2^{ℓ}

Root — n each

Except for the leaves
Caches & Samples empty

for the leaves
the input nodes → Cache
Samples are empty

Stages : $(3 \log n)$
each stage executes in $O(1)$ time
with n processors.

1 to $3 \log n$

t^{th} stage for a node u at level k with v & w as its children

Step 1: Draw samples from Cache of u to form the sample of u

Step 2: Merge the samples (v w) to form the Cache of u

$C_t(u)$: Cache at u after the t^{th} stage

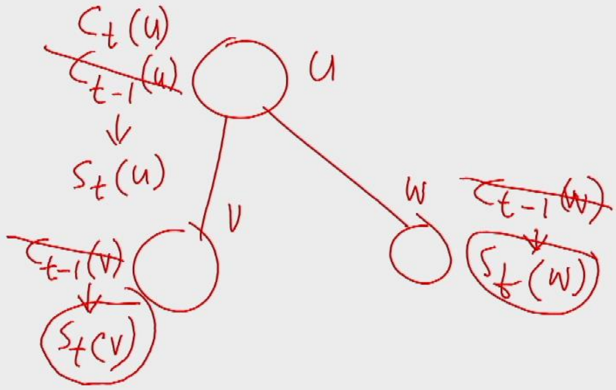
$S_t(u)$: Sample at u ______
______

Stage t

Step 1: Samples $C_{t-1}(u) \longrightarrow S_t(u)$

if $t \in [2k+1, 3k+1]$
then pick every 4^{th} item from RHS
else if $(t == 3k+2)$
pick every 2nd item from RHS
else if $(t == 3k+3)$
pick every item as a sample

Step 2: Merge $S_t(v)$ & $S_t(w)$
to form $C_t(u)$

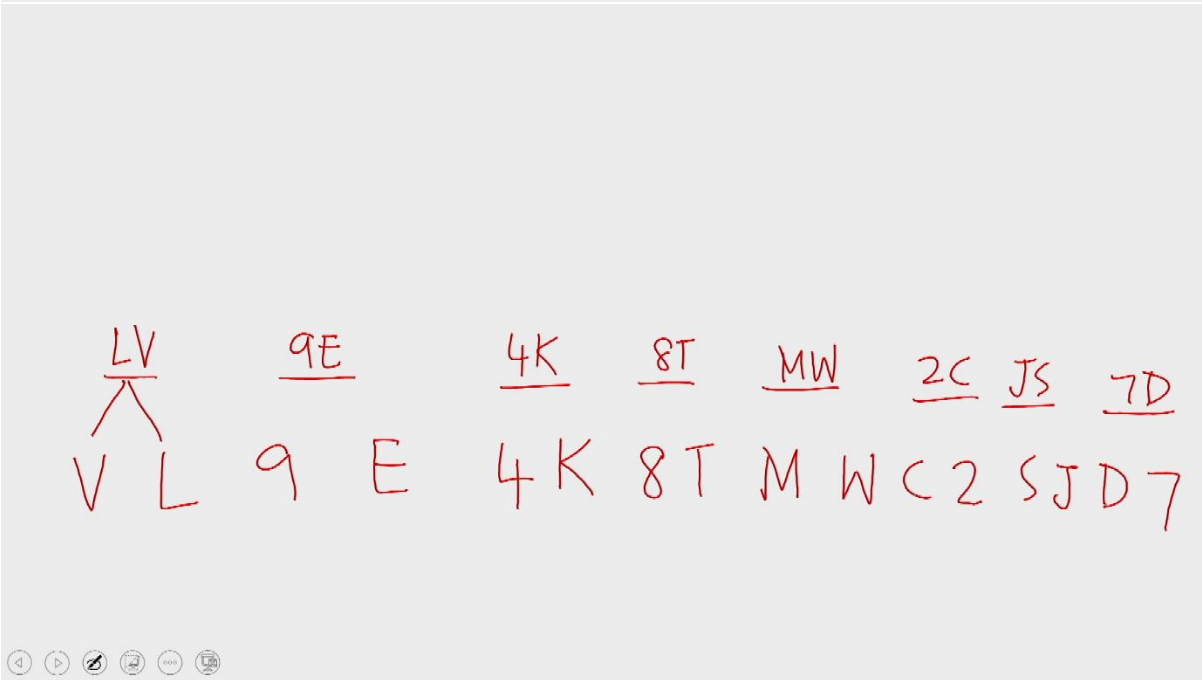


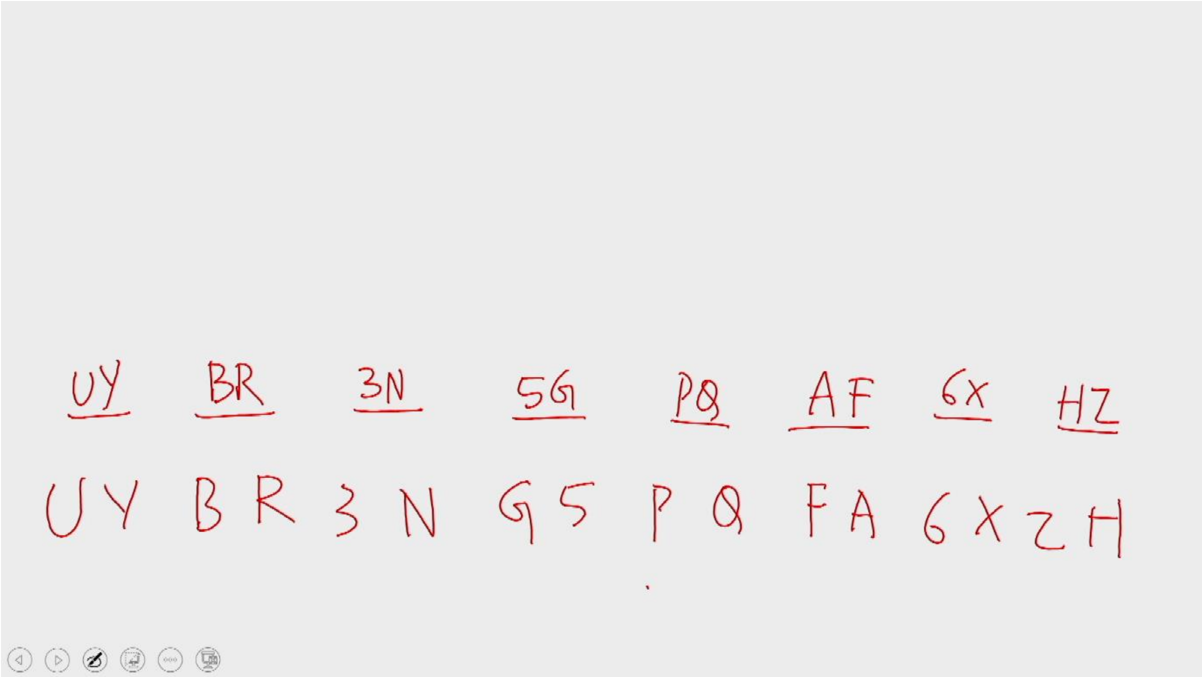
$0 \dots 9 \quad A \dots Z$

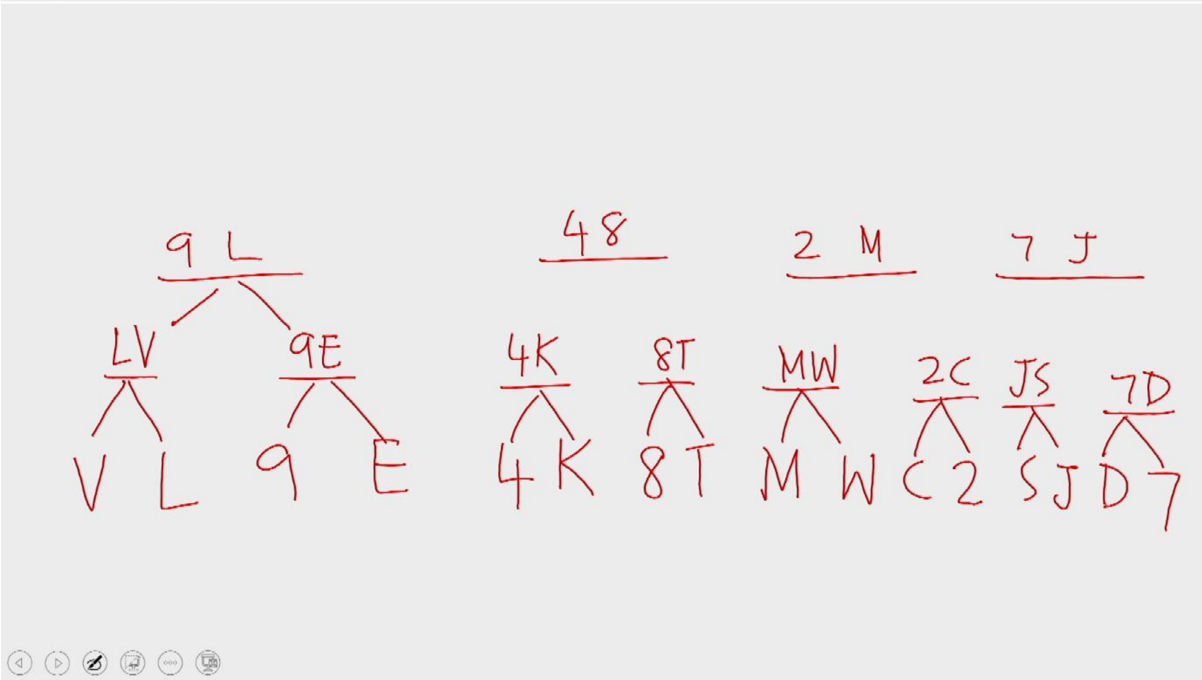
$0 < 1 < \dots < 9 < A < B < \dots < Z$

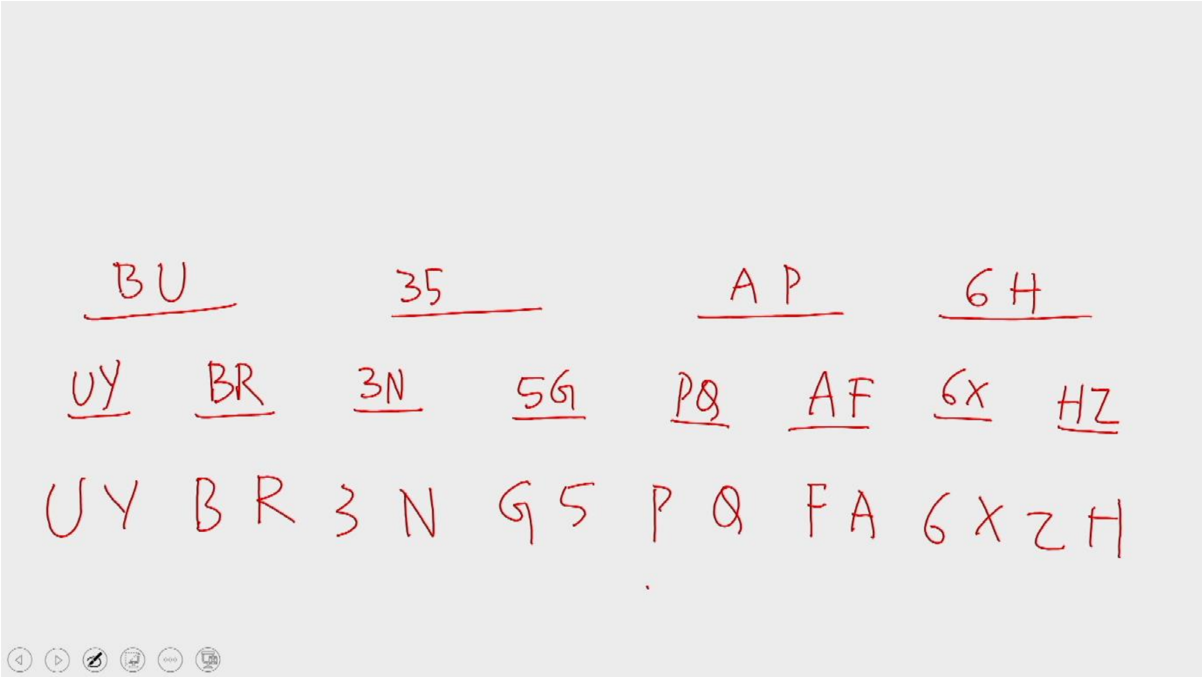
V L 9 E 4K 8T M W C 2 SJD7

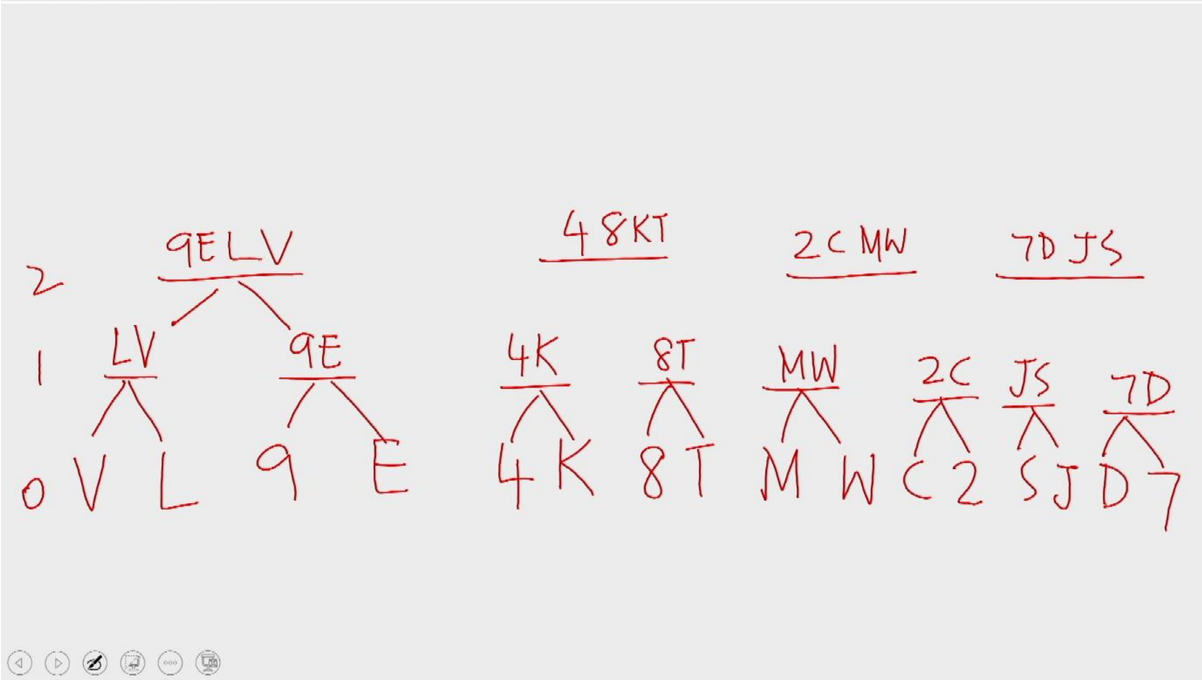

UY BR 3 N G5 PQ FA 6 X Z H

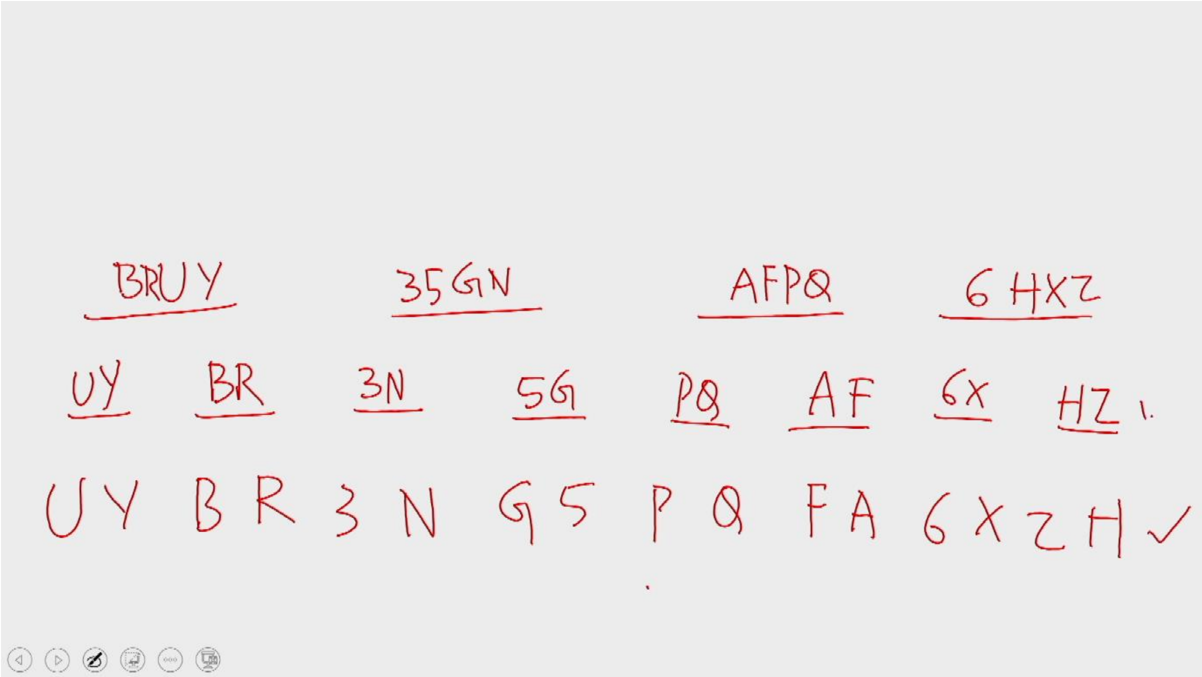

LV 9E 4K 8T MW 2C JS 7D
V L 9 E 4K 8T M W C 2 S J D 7

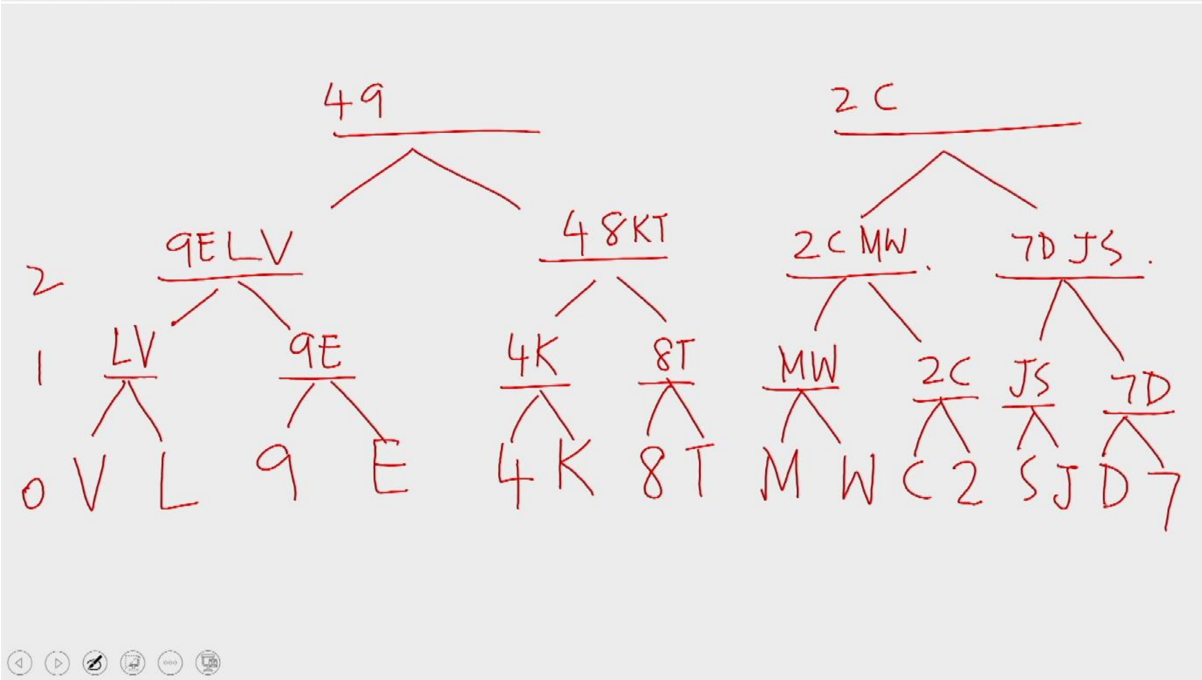

UY BR 3N 5G PQ AF 6X HZ
U Y B R 3 N G 5 P Q F A 6 X Z H

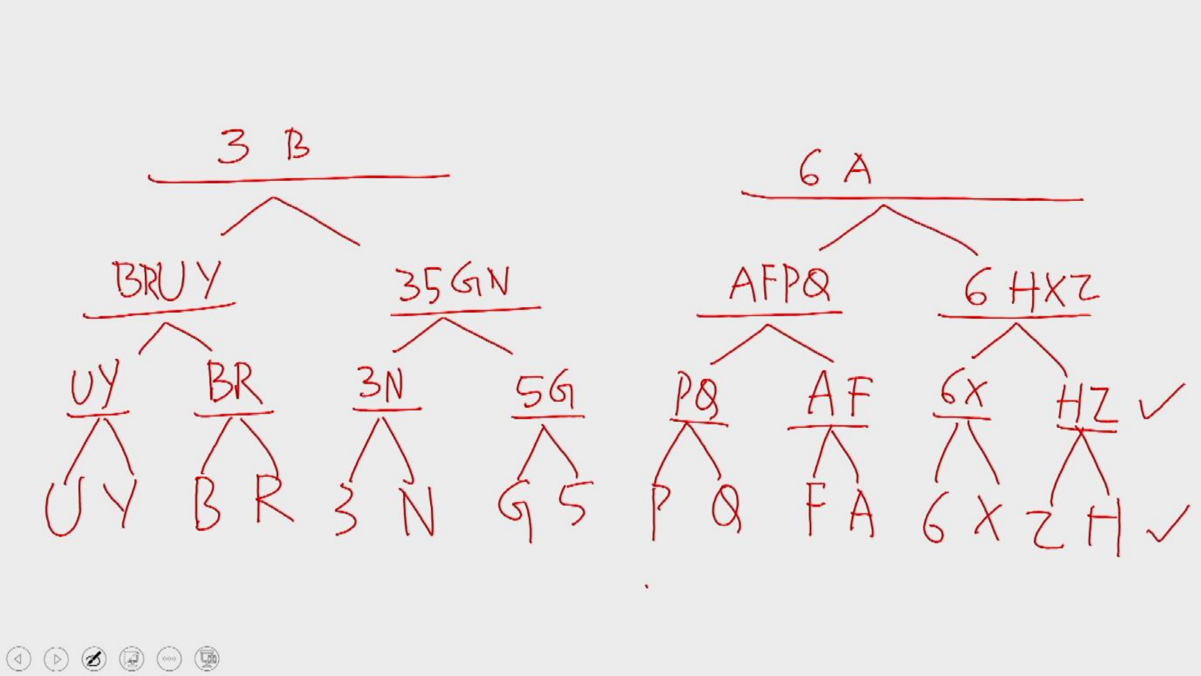

9 L
48
2 M
7 J
LV
9E
4K
8T
MW
2C
JS
7D
V L 9 E 4 K 8 T M W C 2 S J D 7

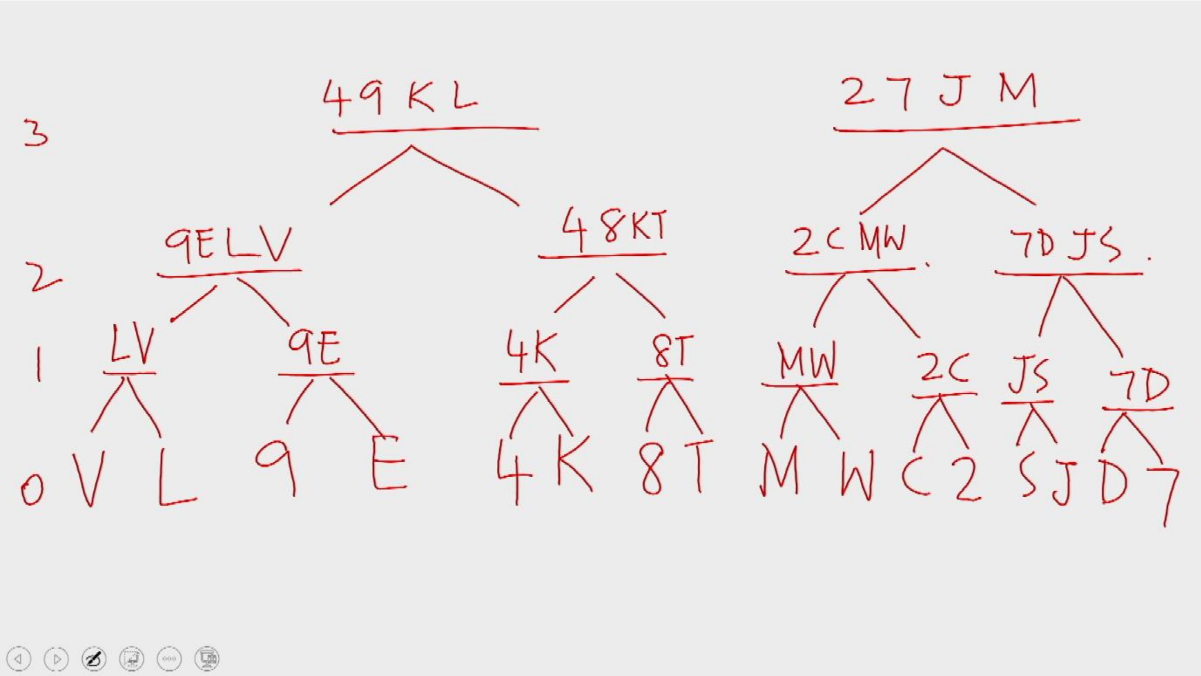

BU 35 AP 6H
UY BR 3N 5G PQ AF 6X HZ
U Y B R 3 N G 5 P Q F A 6 X Z H

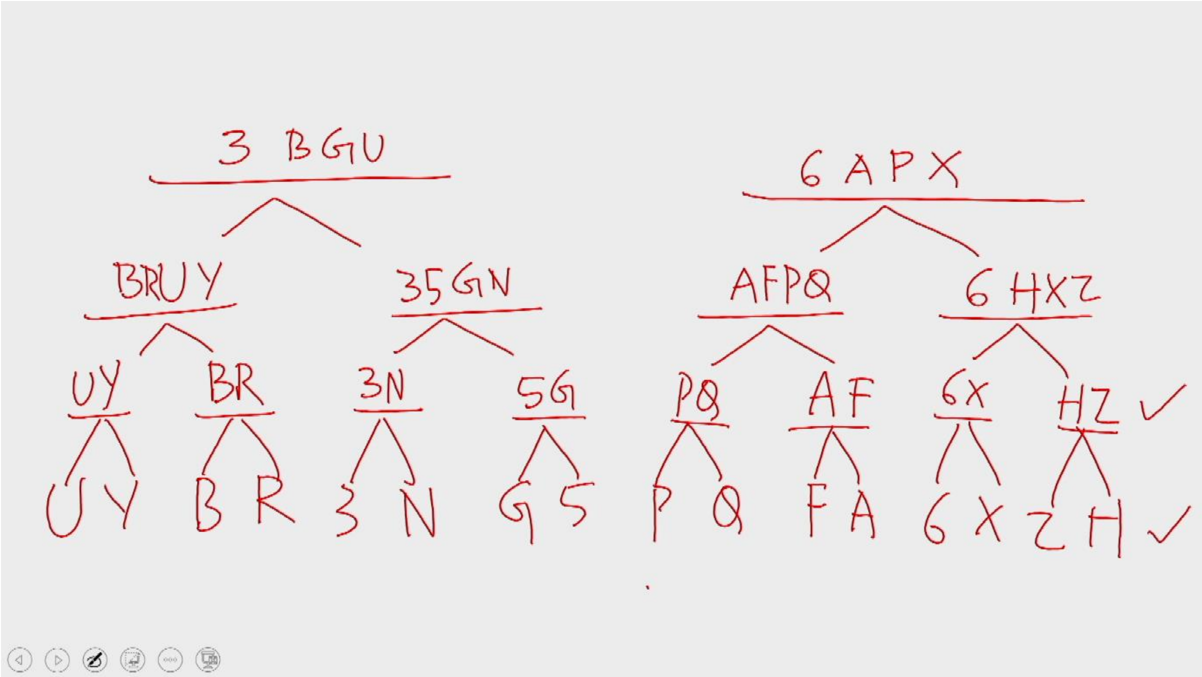

2 9ELV 48KT 2CMW 7DJS
1 LV 9E 4K 8T MW 2C JS 7D
0 V L 9 E 4 K 8 T M W C 2 S J D 7

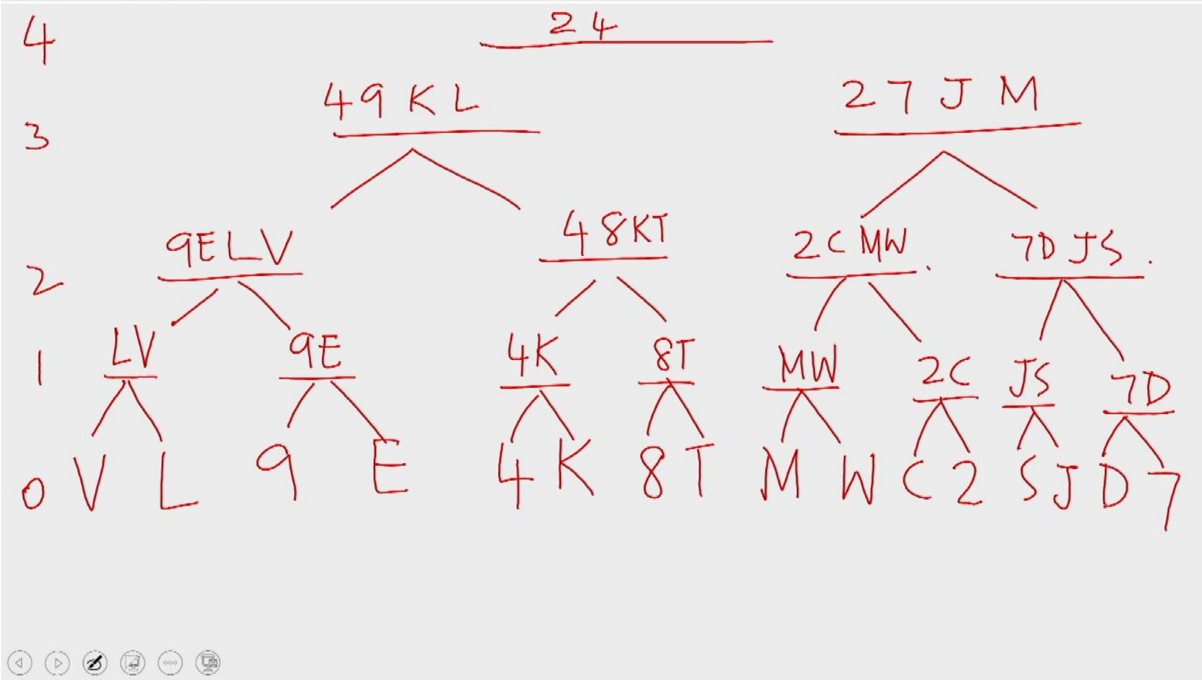

BRUY 35GN AFPQ 6HXZ
UY BR 3N 5G PQ AF 6X HZ
UY BR 3N G5 PQ FA 6X ZH

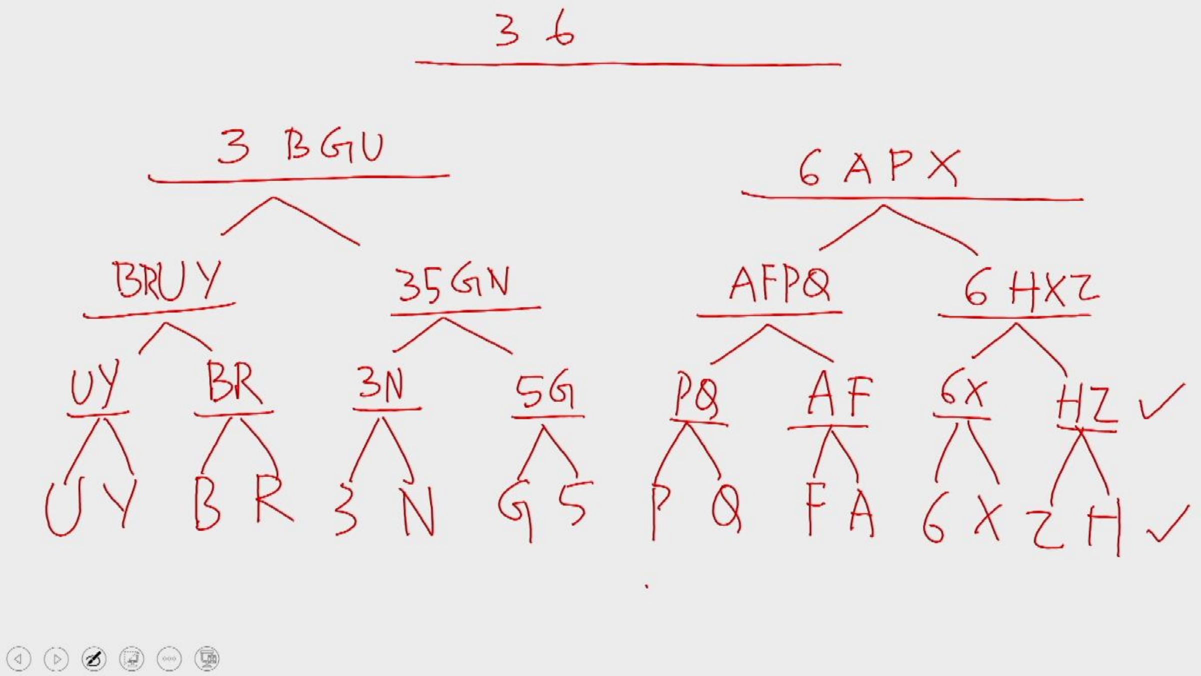

49 2C
9ELV 48KT 2CMW 7DJS
2
LV 9E 4K 8T MW 2C JS 7D
1
0 V L 9 E 4 K 8 T M W C 2 S J D 7

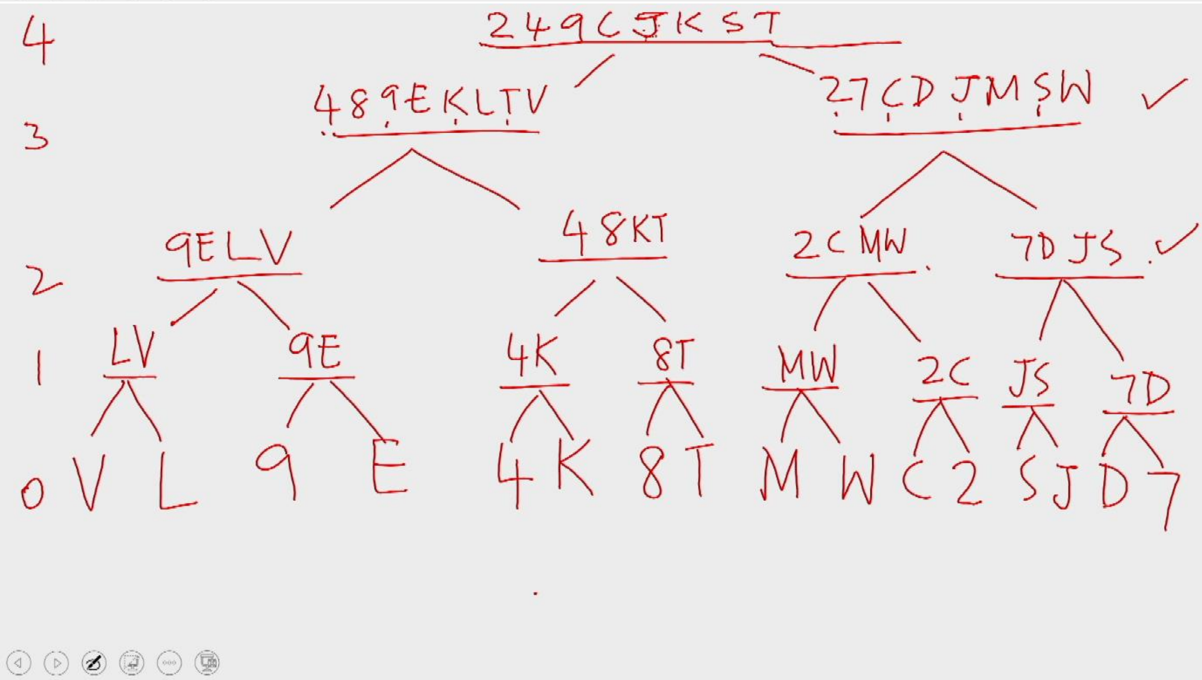

3 B 6 A
BRUY 35GN AFPQ 6HXZ
UY BR 3N 5G PQ AF 6X HZ
U Y B R 3 N G 5 P Q F A 6 X Z H

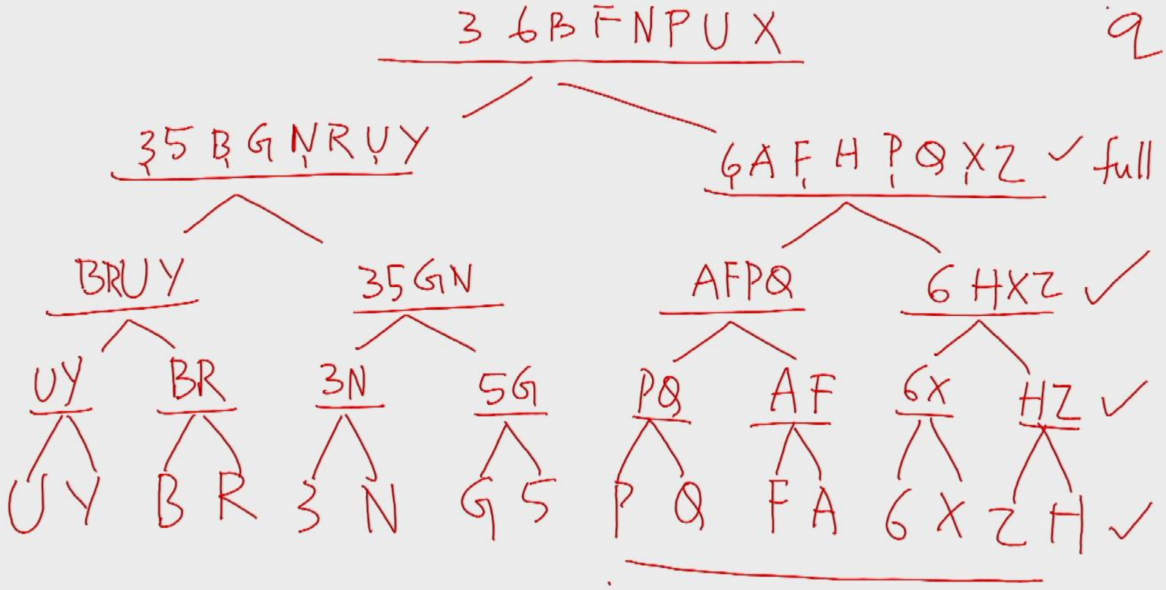

3 49KL 27JM
2 9ELV 48KT 2CMW 7DJS
1 LV 9E 4K 8T MW 2C JS 7D
0 V L 9 E 4 K 8 T M W C 2 S J D 7


3 BGU
6 APX
BRUY
35GN
AFPQ
6 HXZ
UY
BR
3N
5G
PQ
AF
6X
HZ
U Y B R 3 N G 5 P Q F A 6 X Z H


4
24
3
49KL
27JM
2
9ELV
48KT
2CMW
7DJS
1
LV
9E
4K
8T
MW
2C
JS
7D
0
V L 9 E 4 K 8 T M W C 2 S J D 7


3 6
3 BGU
6 APX
BRUY
35GN
AFPQ
6HXZ
UY BR 3N 5G PQ AF 6X HZ
U Y B R 3 N G 5 P Q F A 6 X Z H


4
249CJKST
3
489EKLTV
27CDJMSW
2
9ELV
48KT
2CMW
7DJS
1
LV 9E 4K 8T MW 2C JS 7D
0
V L 9 E 4 K 8 T M W C 2 S J D 7



At the beginning

(end of the hypothetical 0^{th} stage)

leaves are full

Basis

Claim: level k nodes become full at the end of the $3k^{th}$ stage

Hypothesis: Level $k-1$ nodes become full at the end of $(3k-3)^{rd}$ stage

Stage: $3k-2$
level $(k-1)$: 4^{th} element

Stage: $3k-1$
level $(k-1)$: 2^{nd} element

stage: $3k$
level $(k-1)$: every

in stage k

level k nodes get

all inputs in their subtrees

in their caches.

full!

Root is at level $\log n$

becomes full

in stage # $\underline{\underline{3\log n}}$

$3\log n$ stages are enough to

<u>sort</u>

$O(1)$ time stages?

Merging with covers