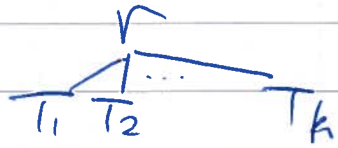
ϕ 3.19

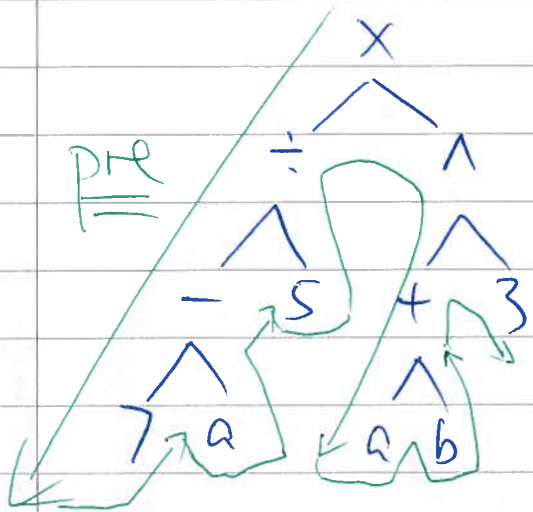
§12.2 Rooted trees

Recall Polish notation

Def. Pre/Post/In traversal



└→ visit r then Pretraversal of T_1, Pret of T_2, ...



$\times \div - 7 a 5 \wedge + a b 3$ pre

$\big((7-a)\div 5\big)\times\big((a+b)\wedge 3\big)$ in

? post

- **Depth First Search** (to find a spanning tree)

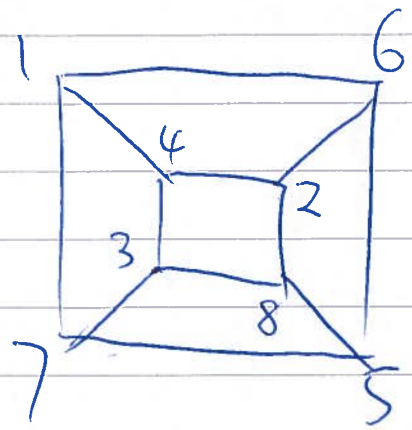
Step
1. $v_1 = v$ $T = \{v_1, \phi\}$ visit v_1
2. find min i s.t. v_i is not visited & $\exists (v_i, v) \in E$ $\begin{cases} \nexists\, i \to 3 \\ \exists\, v_i = v \\ T = T + (v, v_i) \end{cases}$
3. $v = v_1$ 終
4. $v \neq v_1$, backtrack v to parent u & $u = v$ → go to 2.

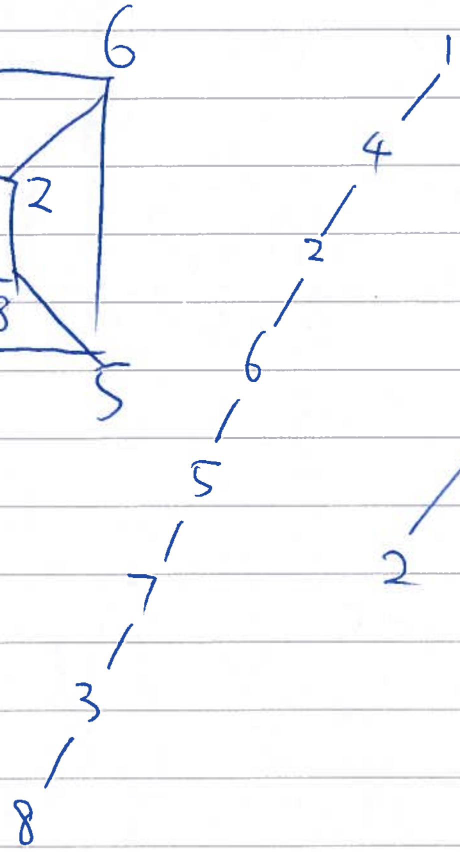
- **Breadth First Search**

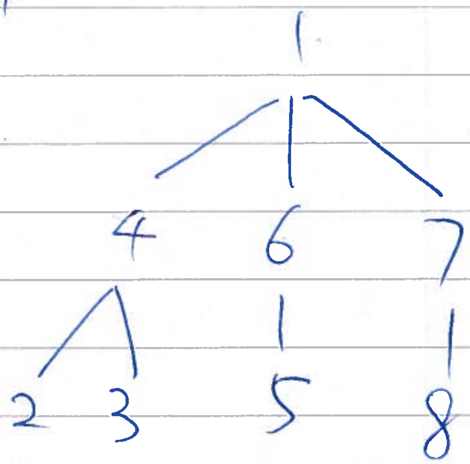
Step
1. $v = v_1$ $T = \{v_1, \phi\}$. $Q = \{v_1\}$ queen
2. If $Q \neq \phi$ delet v from front & add v_i s.t. not visited $\exists (v, v_i) \in E$ to Q & $T = T + (v, v_i)$

2

Ex.







Def. T (V,E) is a m-ary tree if $odeg(v) \le m$ $\forall v \in V$. // $m=2$ binary

If $od(v) \in \{0, m\}$ $\forall v \in V$, then T is called a complete m-ary tree

Thm T is a complete m-ary tree. $|V| = n$

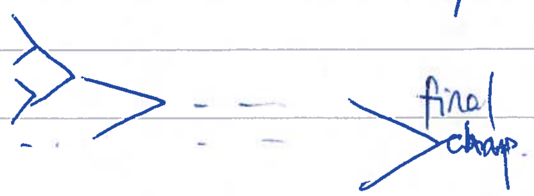
If T has l leaves & i internal vertices

then $n = mi + 1$

$l = (m-1)i + 1$ $\quad (= n - i)$

$i = (l-1)/(m-1) = (n-1)/m$

Ex. Wimbledon tennis championship single-elimination



l = leaves = #players

$m = 2$

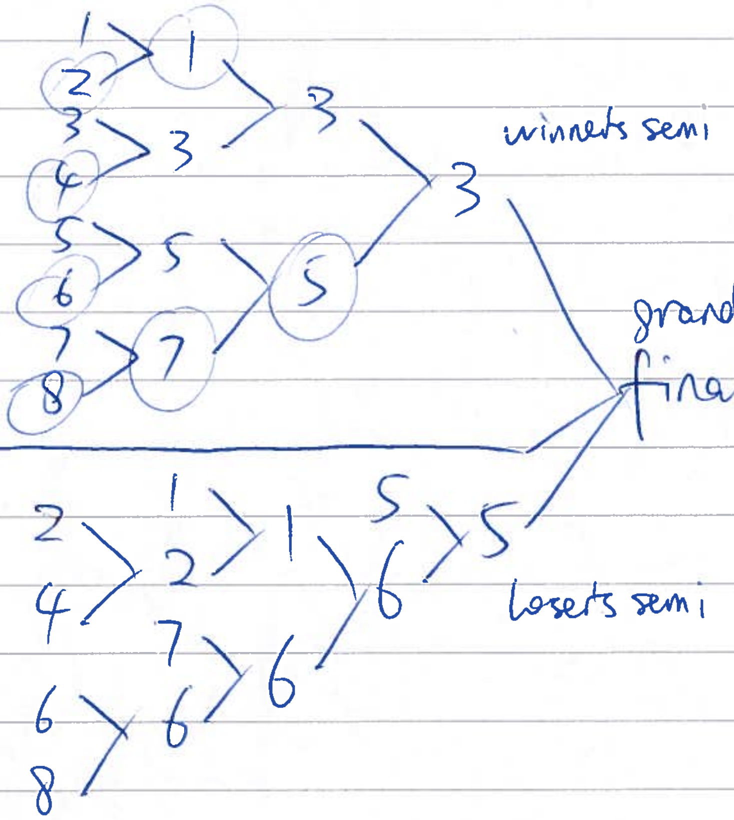
i = maches $= l - 1$

(3)

Ex: Double elimination

$(8-2)\times 2+1$

$13 + \text{final}$
$\quad = 1 \text{ or } 2$



winner's semi

grand final = winner's final + loser's final (if needed)

loser's semi

④

Def $T=(V,E)$ $\forall$ leaf v. level of v = length of path from r to v.
$\max\{\text{levels of leaves}\}$ = height of T $(=:h)$

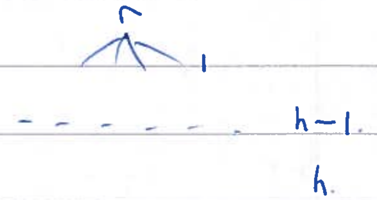
- T is balanced if $\forall$ leaf, its level $= h$ or $h-1$.

Ex. In a single-elimination competition, usual T is balanced
(not counting extra qualification matches)

Thm T: complete m-ary tree of height h
and #leaves $=\ell$.

Then $\ell \le m^h$ & $h \ge \lceil \log_m \ell \rceil$

Cor. $h = \lceil \log_m \ell \rceil$.



$m^{h-1} + m - 1 \le \ell \le m^h$

§12.3 Sorting.

$x_1 \cdots x_n$

$\rightsquigarrow$ (de/ascending order

- Bubble sorting.

$x_1 \cdots\cdots x_n$

For $i=1$ to n

<u>Round 1</u>.

For $j=n$ to $j=i+1=2$
compare x_j & x_{j-1}

<u>Round 2</u>.

For $j=n$ to $j=3$.

7 9 2 5 8

7 2 9 5 8

2 7 9 5 8.

end up smallest number "bubble" out.

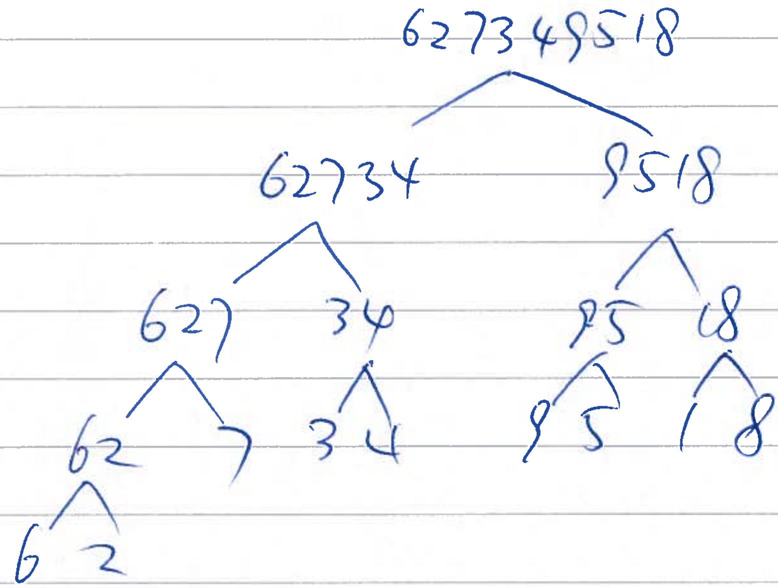
<u>Complexity</u> $\binom{n}{2} = \frac{n(n-1)}{2}$ comparisons in worst-case

⑤

- Merge Sort $x_1 \cdots x_n$

Step1 Splitting n items into $\lceil \frac{n}{2} \rceil$ & $\lfloor \frac{n}{2} \rfloor$

& keep going until all lists have 1 item



$L_1 = 6$

$L_2 = 2$

$L_3 = 7$

$\vdots$

Step2 Operation: Merge L_i & L_j.

M: $L_i \quad L_j = L_{ij}$

1° Set $L_{ij} = \phi$

2° Compare first elements of L_i & L_j

$L_{ij} = L_{ij} + \min$ of them & $L_i (\text{or } L_j) = L_i (\text{or } L_j) - \text{this min}$

3°. If $L_i = L_j = \phi$ $\boxed{/\!/}$

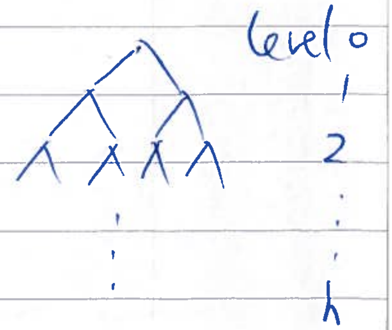
$L_i = \phi \quad L = L + L_j$

$L_j = \phi \quad L = L + L_i$

otherwise back to 2.

- Worse-case time complexity.

"WLOG" $n = 2^h$



$$\#\text{comparisons} = 2^{h-1} + 2^{h-2}(2-1) + \cdots$$

$$= \sum_{k=1}^{h} 2^{k-1}(2^{h-k+1} - 1) = \sum_{k=0}^{h-1} 2^h - 2^k = h \cdot 2^h - (2^h - 1)$$

$$= n \log_2 n - n + 1$$