§12.2 Rooted trees.

Recall Polish notation

Def. Pre/Post/In-traversal

\[ T_1 \rightarrow \ldots \rightarrow T_k \]

\[ \text{pre} \rightarrow \text{visit } r \text{ than Pre-traversal of } T_1 \text{, Pret of } T_2, \ldots \]

\[ X \xrightarrow{\text{pre}} 7 \xrightarrow{\text{pre}} 5 \xrightarrow{\text{pre}} \frac{85}{3} a + ab \]

\[ (\frac{7-a}{3}) \cdot (a+b)^3 \]

in

post

• Depth First Search (to find a spanning tree)

Step 1. \( V_i = V \rightarrow T = T \cup V_i, \emptyset \rightarrow \) visit \( V_i \),

2. find min \( i \) s.t. \( V_i \) is not visited \& \( \exists (V_i, V) \in E \)

3. \( V = V_i \rightarrow \)

4. \( V \notin V_i \), backtrack \( V \) to parent \( u \) \& \( u = V \rightarrow \) go to 2.

• Breadth First Search

Step 1. \( V = V_i \rightarrow T = T \cup V_i, \emptyset \rightarrow \) Queen

2. If \( Q \neq \emptyset \), delet \( V \) from front \& add

\( V_i \) s.t. not visited \( \exists (V_i, V) \in E \rightarrow \) to \( Q \)

\& \( T = T \cup (V_i, V) \)
Def. $T$ is a $m$-ary tree if $\forall v \in V, d_v \leq m$ // $m=2$ binary

If $d_v(v) < m$, then $T$ is called a complete $m$-ary tree $\forall v \in V$.

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

If $T$ has $L$ leaves & $i$ internal vertices
then $n = m^i + 1$
$L = (m-1)i + 1 = n - i$
$i = (L-1)/(m-1) = n-1/m$.

Ex. Wimbledon tennis championship single-elimination

$\Rightarrow$ final

$m = 2$
$n = L = \#\text{players}$
$i = \#\text{matches} = L - 1$. 
Ex. Double elimination

\[(8-2) \times 2 + 1\]

Winners' Semi

Grand Final

Final = Winners' Final + Losers' Final (if needed)
Def: \( T = (V, E) \) a leaf \( v \). Level of \( v \) = length of path from \( r \) to \( v \).
\begin{align*}
\text{max \{levels of leaves\}} &= \text{height of } T \ (\equiv h) \\
T \text{ is balanced if } &\forall \text{ leaf, its level } = h \text{ or } h - 1
\end{align*}

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

Thm: \( T \): complete \( m \)-ary tree of height \( h \)
\[ \text{then } l \leq m^h \quad \text{and } \#\text{leaves} = l \]
\[ h = \lceil \log_m l \rceil \]

Cor: 
\[ m^{h-1} + m - 1 \leq l \leq m^h \]

§12.3 Sorting: \( x_1 \ldots x_n \)
\[ \rightarrow \text{descending order} \]

- Bubble sorting

\[ \text{For } i = 1 \text{ to } n \] \[ \text{Round 1:} \]
\[ \text{For } j = n \text{ to } j = i + 1 = 2 \]
\[ \text{compare } x_j \text{ & } x_{j-1} \]
\[ \text{Round 2:} \]
\[ \text{For } j = n \text{ to } j = 3 \]
\[ \text{end up smallest number "bubble" out} \]

Complexity: \[ \binom{n}{2} = \frac{n(n-1)}{2} \] comparisons in worst-case
Merge Sort $x_1, \ldots, x_n$

Step 1: Splitting $n$ items into $\lceil \frac{n}{2} \rceil$ & $\lfloor \frac{n}{2} \rfloor$

Keep going until all lists have 1 item

$6273, 49518$

$L_1 = 6$

$L_2 = 2$

$L_3 = 7$

Step 2: Operation: Merge $L_i$ & $L_j$

1. Set $L_{ij} = \emptyset$

2. Compare first elements of $L_i$ & $L_j$

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

3. If $L_i = \emptyset$, $L_j = \emptyset$

$L = L + L_j$

$L_j = \emptyset$

$L = L + L_i$

otherwise back to 2.

Worse-case time complexity:

"WLOG" $n = 2^h$

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

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

$= n \log_2 n - n + 1$