Proofs

Daniel F. Santos B.

June 14, 2019

Definition 1. A binary tree is a tree in which every node has at most two children.

Height of a Binary Tree is $Log(n)$

Suppose we have A prefect binary tree (PBT), PBT is binary tree in which all internal nodes have exactly two children and all leaves are at the same level.

Proof. Let n be the number of nodes in the PBT and $A(k)$ be the number of nodes in the level k

$$A(k) = \begin{cases} if \quad k = 0 & 1 \\ if \quad k > 0 & 2 * A(k-1) \end{cases} \tag{1}$$

so, given the Equation 1, we can say that the total number of nodes n in a PBT is

$$n = 2^0 + 2^1 + 2^2 + ... + 2^h$$

from Computer Science course we know

$$1 + 2^1 + 2^2 + ... + 2^h = 2^{h+1} - 1$$

therefore:

$$n + 1 = 2^{h+1}$$

$$\begin{aligned} log_2(n+1) &= log_2(2^{h+1}) \\ &= h + 1 \\ h &= log_2(n+1) - 1 \end{aligned} \tag{2}$$

Therefore h is $O(log(n))$

Sorting lower bound in comparision model

Can we say that is possible to sort faster than $\Omega(nlogn)$?, let's suppose we have a decision tree to sort elements, e.g Fig 1 with tree elements

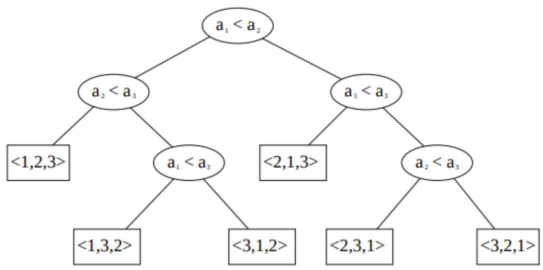

Figure 1: Decision tree with tree elements, taken from Bowdoin College

□

so,each leave will have a permutation of the original array

Proof.

$$
\begin{aligned}
2^h \leq n! \implies h &\geq log_2(n!) \\
&= log_2(n(n-1)(n-2)...(2)) \\
&= logn + log(n-1) + log(n-2) + ... + log2 \\
&= \sum_{i=2}^{n} log(i) \\
&= \sum_{i=2}^{n/2-1} log(i) + \sum_{i=n/2}^{n} log(i) \\
&\geq= 0 + \sum_{i=n/2}^{n} log(\frac{n}{2}) \\
&= \frac{n}{2} log(\frac{n}{2}) \\
&= \Omega(nlog(n))
\end{aligned} \tag{3}
$$

□