2/5 Comparison Sort Bound

Sunday, February 11, 2018 3:22 PM

There are a lot of average case $O(n \log n)$ sorts is there a way to make a sort that is much more efficient in the worst case?

Short answer

No

Long answer

Not using comparison methods

Consider an array of 3 elements

the possible orderings of these elements is

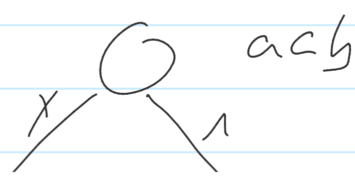
123 132

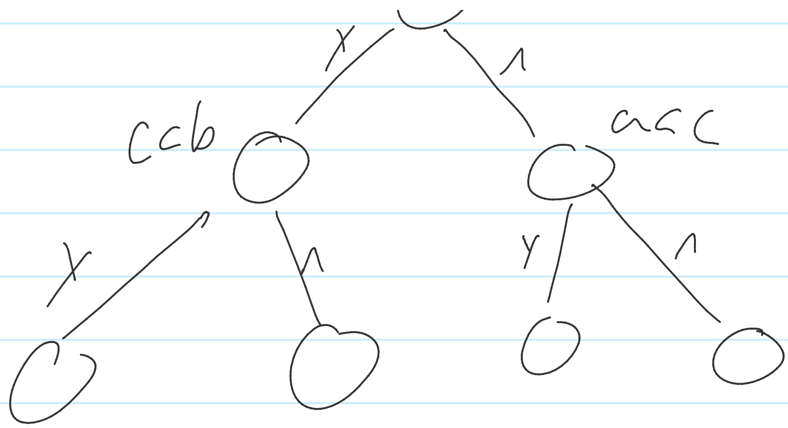
213 231

312 321

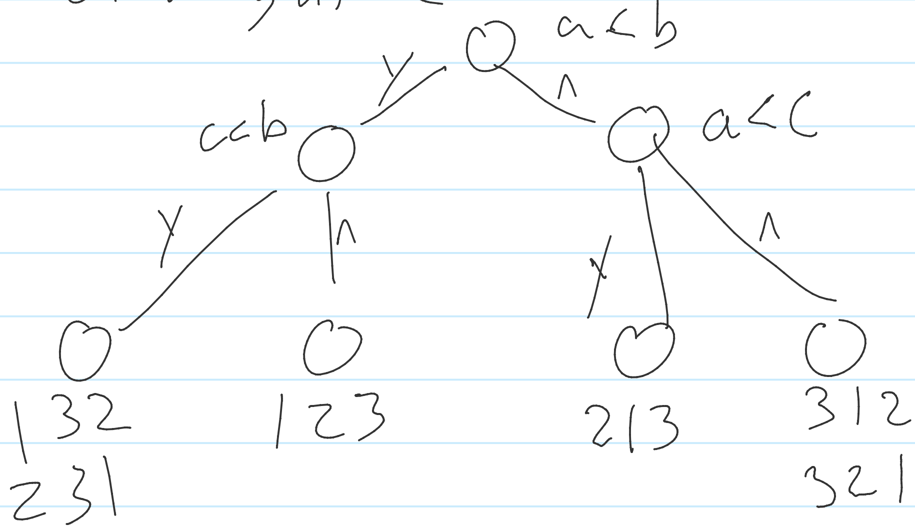
6 orderings

each comparison can lead to two unique possibilities





Consider where each of these permutations would end up with this comparison tree of height 2



By pigeon hole principle regardless of the tree same node has atleast 2 permutations.

Thus to know the permutation to sort the array we must have a tree with atleast depth 3.

with atleast depth 3.

By using this we know that to sort n elements we need atleast a tree with depth $\lceil \log_2(n!) \rceil$

How much is $\lceil \log_2(n!) \rceil$?

$$\log_2(n!) = \log_2(n) + \log_2(n-1) + \cdots + \log_2(1)$$

$$\geq \log_2(n) + \log(n-1) + \cdots + \log\left(\frac{n}{2}\right)$$

$$\geq \log_2(n/2) + \log(n/2) + \cdots + \log(n/2)$$

$$= n/2\left(\log_2(n/2)\right)$$

$$= n/2\left(\log_2(n) - 1\right)$$

which is $\in \Omega(n \log(n))$

So the worst case of any comparison based sort is $\in \Omega(n \log(n))$