Subject : 2012

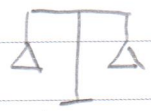
No. : 11-1

Date : ...../...../.....

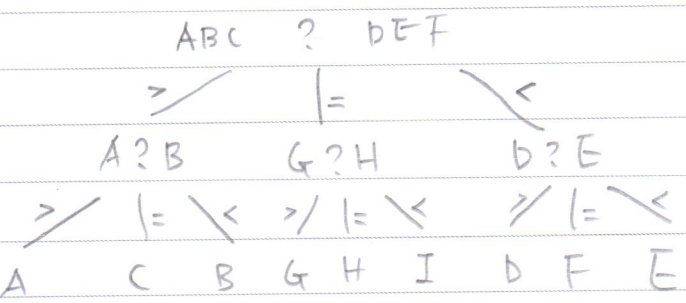
* balance game & trees

9 coins A B C D E F G H I

one of them heavier

two uses of balance 

how to find it ?



a complete trinary tree !

leaf : outcomes

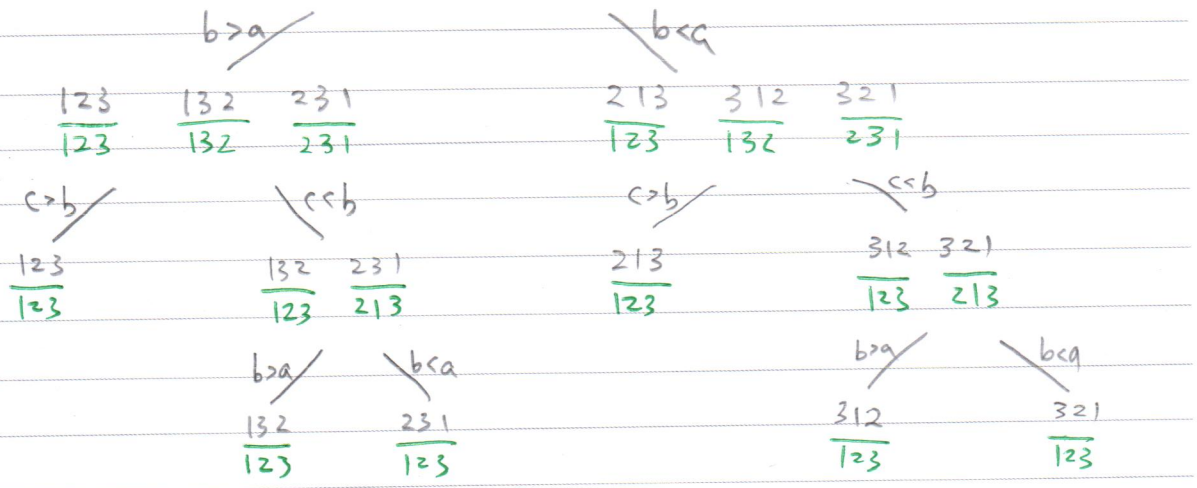
non-leaf : conditions

* two uses of balance = two non-leaf levels

⇒ at most 9 possibilities

* insertion sort (with neighbor swapping)

abc = { 123 or 132 or 213 or 231 or 312 or 321 }



3 comparisons

Double A

* comparison sort

leaf : permutation of original sequence

node : comparisons

#leaves $\geq$ # permutations to make sorting correct

$=$

$n!$

$\Rightarrow$ depth $\geq \log_2(n!) + 1 = \Omega(n \log n)$

$\Rightarrow$ # comparison is $\Omega(n \log n)$

* # Comparison in quick sort

$$C_{avg}(n) = (n-1) + \frac{1}{n} \sum_{i=1}^{n} \left(C_{avg}(i-1) + C_{avg}(n-i) \right)$$

$$= (n-1) + \frac{2}{n} \sum_{i=0}^{n-1} \left(C_{avg}(i) \right)$$

$$C_{avg}(n-1) = (n-2) + \frac{2}{n-1} \sum_{i=0}^{n-2} \left(C_{avg}(i) \right)$$

$$n\,C_{avg}(n) - (n-1)\,C_{avg}(n-1) = n(n-1) - (n-1)(n-2) + 2\,C_{avg}(n-1)$$

$$\Rightarrow n\,C_{avg}(n) = (n+1)\,C_{avg}(n-1) + 2n - 2$$

$$\Rightarrow \frac{C_{avg}(n)}{n+1} = \frac{C_{avg}(n-1)}{n} + \frac{2}{n} - \frac{4}{n(n+1)}$$

$$= \frac{C_{avg}(0)}{1} + \underbrace{\sum_{i=1}^{n} \frac{2}{i}}_{O(\log n)} - \sum_{i=1}^{n} \frac{4}{i(i+1)}$$

$$\Rightarrow C_{avg}(n) = O(n \log n)$$

* non-comparison sort

counting sort for keys in $\{1, 2, \cdots, K\}$

① construct an array C of size K (O(K) space, O(1) time)
② count (# keys = k) and store in C[k] (O(n) time)
③ compute start[k] = $\sum_{i=1}^{k-1} C[i]$ (O(K) space, O(K) time)
④ put element a[j] in position start[a[i]] + □
(O(n) time)

⇒ O(n+K) time, O(K) space

* sorting integers

179, 208, 306, 93, 859, 984, 55, 9, 271, 33

① compare by hundreds

984
859
306
208, 271
179
093, 055, 009, 033

② compare by tens

208, 271
009, 033, 055, 093
⋮

④ final result

984, 859, 306, 208, 271, 179,
009, 033, 055, 093

most-significant digit (MSD) sort

(complicated to implement)

Subject : 2012

No. : 11-4

Date : ...../...../.....

* least-significant digit first ?

271		306		009
093		208		033
033		009		055
984	stable ⇒	033	stable ⇒	093
055		055		179
306		859		208
208		271		271
179		179		306
859		984		859
009		093		984
sorted		sorted		sorted

LSD radix sort

MSD radix sort

* bin/ bucket sort:

similar to counting but maintain elements in each bin