DSA - A.

3 (a) we divide the 100 programmes into 10 batches of 10 each and conduct ~~a~~ session with each batch and there by using 10 sessions up to now, we take the top in each and we get 10 top 1 ranks each from each group and now conduct another session (11th) to find the best programme.

(b) conducting 10 sessions we eliminate 9 from each session there by eliminating at max 90 = 9 x 10 eliminated twice

there by still 10 coders are left that havent been eliminated for the best programe

∴ hence not possible in 10 comparisions

(c) similar to (a) we conduct 10 sessions in 10 batches of 10 each and now the 11th session is used b/w the 10 top rankers from each batch we get the 1st position from here

no we consider all the possible cadetes for 2nd, 3rd and 4th places

the top 4 from 11th session are possible candidates now considering these top 4 from 11th session the 1st from 1, the could have defeated 2nd, 3rd and 4th rankers of the tournament in his batch.

hence we consider top 4 from (1st of 11th)'s batch.

the 2nd of 11th could have defeated 3rd and 4th rankers of the tournament in his batch hence we consider top 3 from (2nd of 11th)'s batch.

similarly
top 2 from (3rd of 11th)'s batch
and top 1 from (4th of 11th)'s batch

$4+3+2+1=10$

arranging another match between these 10 (12th)

would give me the 1st, 2nd, 3rd and 4th ranks of the contest

$\therefore$ hence done in 12 sessions.

2) We follow the below logic

first take last 4 (b, c, d, e) and find min (step1)

in 3 comparisions and eliminate it

whatever we compered with the eliminated one in its first comparision, now compare it with "a" and find min of these 4

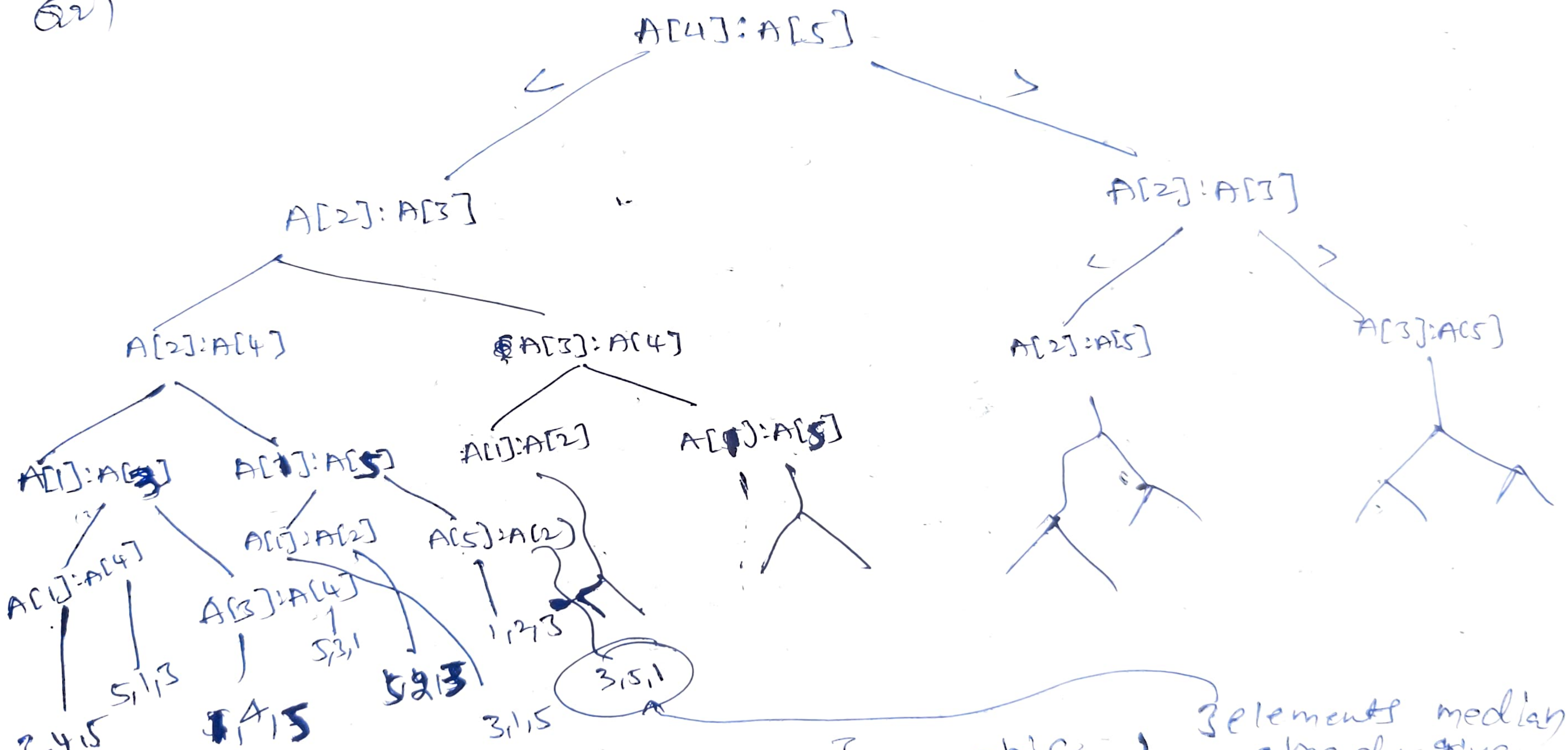
this takes only 2 comparisions as we already know one of the comparisions result from previous 3 comparisions, we eliminate this min too

with the 3 left we again find the min step 3 and this would be the medien

this takes only 1 comparisions as we alreday know the result of one of the required comparisions from step1 (or) step2

3+2+1 = 6 comparisions

Q2)



3 elements median already given in pdf we can use it

Similarly others are Isomorphic

(a) 1

Given

value Arr, weight array

Algorithm for MoM

The medians of median finds the largest element in an array in the time complexity of O(n) It follows the fact that sum of O(n)(finite) is also O(n) complexity.

we will eliminate some of the elements, after dividing the elements in groups of 5. we will eliminate 30% of the elements the why recursively do this for finding the median.

approach for problem

our task is to find the weighted median for this value array and weight array.

for this I will find median in the value array I will arrange all the elements left to it as its lower value and every thing higher to it in right side

Then i will calculate the left weight and right weight if both are lesser than the half of total weight.

then the number we are at is the required number.

if the left sum is greater than $\frac{w}{2}$ (w = total weight) now apply the same procedure for all the elements which are left to if it's the right sum is greater than $\frac{w}{2}$ → follow the same for the elements in the right.

by doing this recursively we can find the weighted medians

we will not get both weights greater than $\frac{w}{2}$ because if we add both left and Right sum it must be lesser than w.

we know that the complexity ~~for~~ for finding MOM is $O(n)$ finding the sum and right sum takes the complexity of $O(n)$ after every step we will eliminate half the elements that are present at the previous step.

$$= cn + n + \frac{cn}{2} + \frac{n}{2} + \cdots$$

$$= 2cn + 2$$

If we go towards left then ~~we~~ we will find sum of elements b/w the present median and previous median we will sub stract it from leftsum and add it righ sum and similarly, for right sum

The cn is for median of med and the n is for calculating the sum b/s the new domain