ID: 21201240

Ans to Ques no 1 (a)

As the award is given in a constant time. The whold award given time complexity would be $O(n)$.

Ans to the Ques no 1 (b)

The asymptotic time complexity of the following function is:

$$O(1) * O(n) * O(n) * (O(1) + O(1))$$

$$\Rightarrow O(n^2)$$

Ans to the Ques no 1 (c)

Following function is:

$$T(n) = 625T\left(\frac{n}{5}\right) + n^3$$

As per Master's Theorm,

$$T(n) = aT\left(\frac{n}{b}\right) + cn^k$$

here,

$a = 625$

$b = 5$

$k = 3$

$c = 1$

ID: 21201290

$\therefore b^k < a$

$\Rightarrow 5^3 < 625$

$\Rightarrow 125 < 625$

if $b^k < a$, the time complexity would be $O(n^{\log_b a})$

$\therefore T_c = O(n^{\log_5 625})$

$= O(n^4)$

Ans

ID: 21201840

Ans to the Ques no 2 (a)

Time complexity of the following function is:

$$O(\log_7 n) * O(n) * (O(n)+O(n))$$

$$\Rightarrow O(n^2 \log_7 n)$$

Ans the Ques no 2 (b)

Given recurrence relation is:

$$T(n) = T(n/2) + T(n/4) + n$$

Substitution method,

$$T(n) = T(n/2) + T(n/4) + n$$

$$= \left[T(n/2^2) + T(n/2^3) + n/2\right] + n$$

$$= \left[T(n/2^3) + T(n/2^4) + n/2^2\right] + n/2 + n$$

$$= \left[T(n/2^4) + T(n/2^5) + n/2^3\right] + n/2^2 + n/2 + n$$

$$= T(n/2^k) + T(n/2^{k-1}) + (n/2^{k-2}) + n/2^{k-3} + n/2^{k-4} + n$$

ID: 21201240

Assume,

~~n~~ 2^k

$n/2^k = 1$

$n = 2^k$

$k = \log_2 n$

$\therefore$ Time complexity of the given recurrence relation is $O(\log n)$

Ans

Sorting 1 :

a) Since there is no requirement about stability and it should need to do it in place method that's why I choose to bubble sort.
In bubble sort we will move left to right swapping adjacent elements as needed. Each pass moves the next largest element into its final position

15	8	11	112	33	39	88	41

↓

8	15	11	112	33	39	88	41

↓

8	11	15	112	33	39	88	41

↓

8	11	15	33	112	39	88	41

↓

8	11	15	33	39	112	88	41

↓

8	11	15	33	39	88	112	41

↓

8	11	15	33	39	88	41	112

↓

8	11	15	33	39	41	88	112

In bubble sort it iterates each elem and compare with the next one: if $a[i] > a[i+1]$ then it'll swap otherwise $a[i] == a[i+1]$.

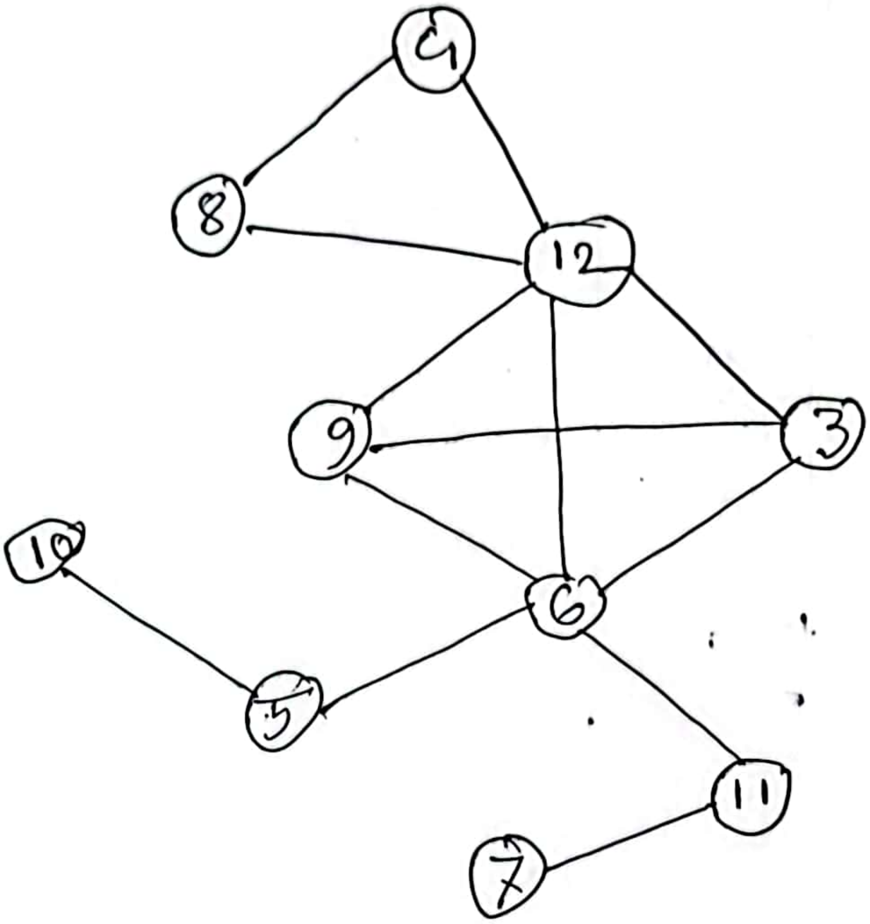
b) The best case occurs when the array is already sorted. In a sorted list there is 0 number of swaping and $(n-1)$ numbers of comparison. Hence the best case of time complexity of bubble sort is $O(n)$.

Ans.

④ 21201240

Graph 2:

a)



b) Yes bill is right. there are atleast 4 triangle.

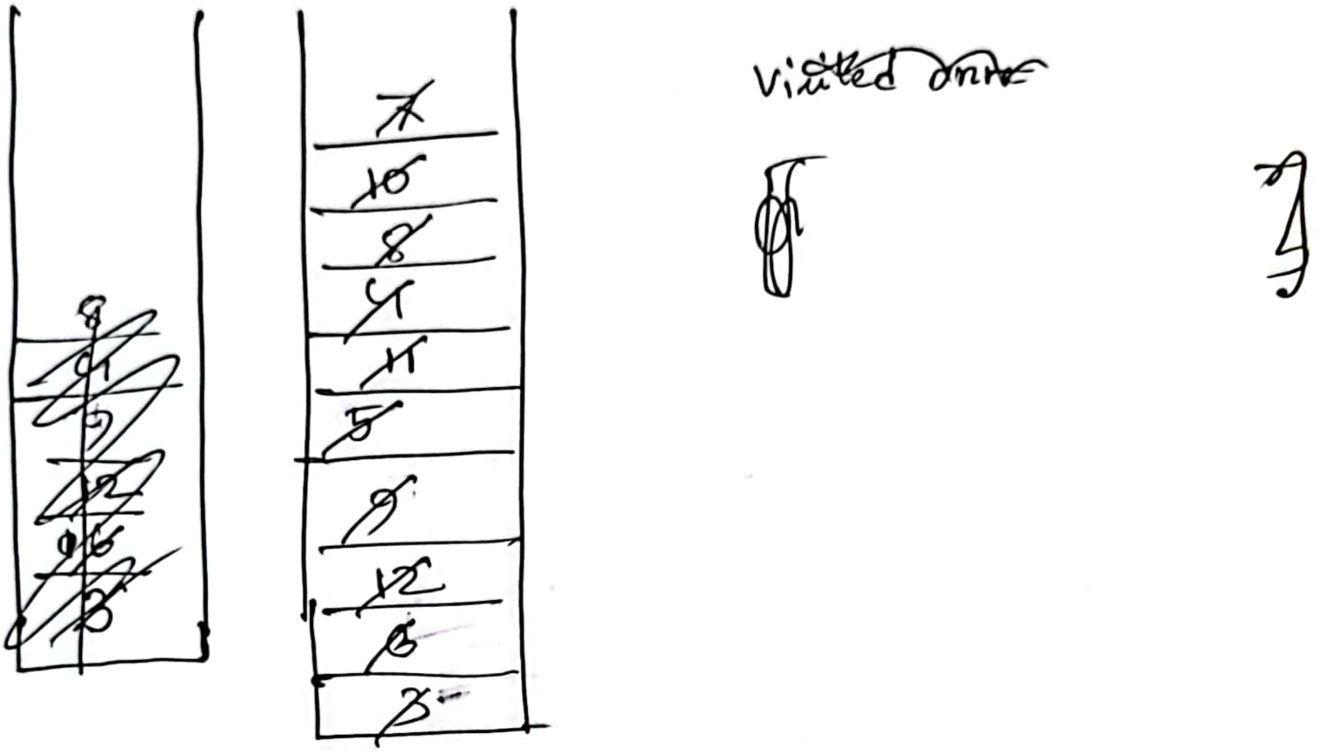
we can see from the graph.

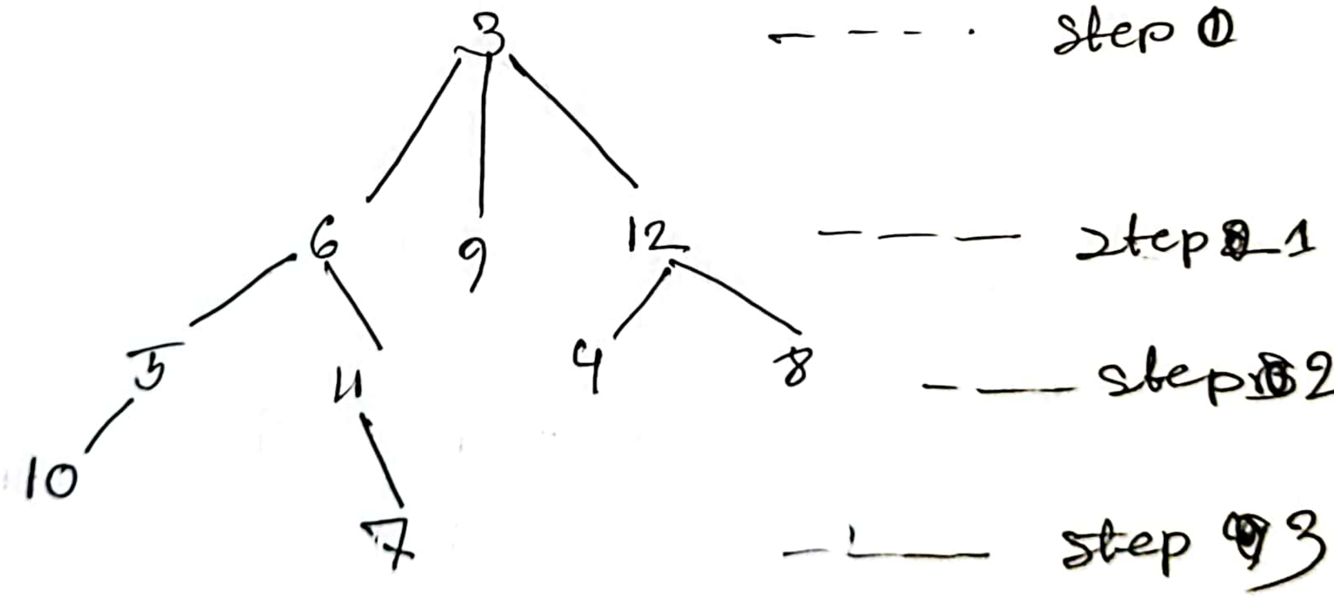
edge [4,8,12], [3,9,12], [3,6,9], [12,9,6]

~~nodes a~~ are connected and make triangles.

21201240

c) From the graph





for node [6,9,12] shortest distance is 1 from node 3 and for [5,11,4,8] shortest distance is 2 from node 3 and for [10,7] the shortest distance is 3 from node 3

d) In this graph we saw that we have 13 edges. From no 3 to 12 vertex we have $\frac{n(n-1)}{2}$ number of edges if we keep the graph simple. Now, for 3 to 12 number of nodes the total edges will be = $\frac{10(9)}{2}$

$= 45$ edges

$\therefore$ Number of edges we need more

$= (45-13)$

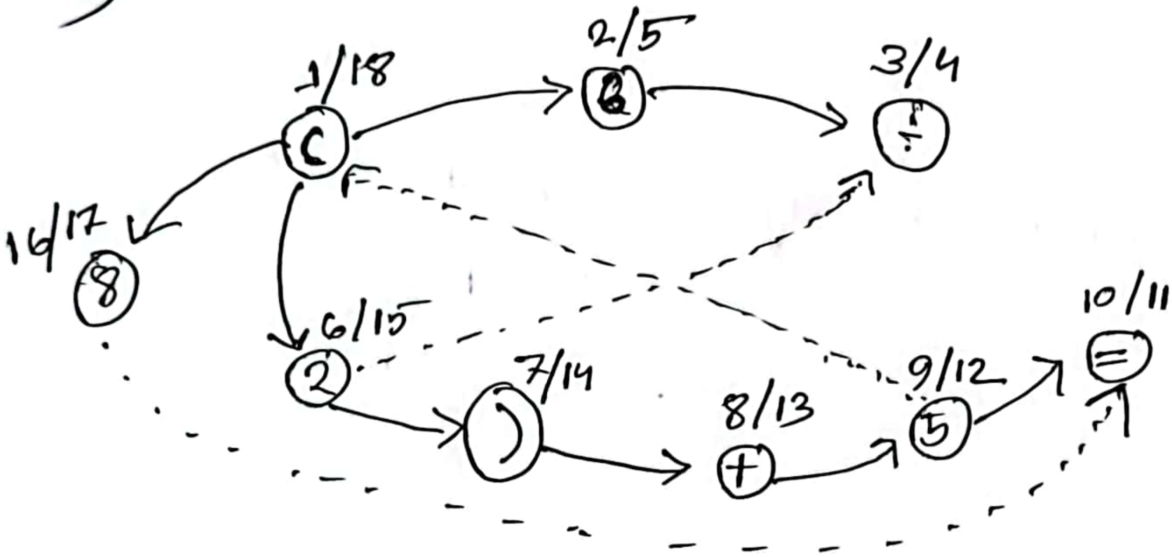
$= 32$ edges Ans

⑩ 21201240

Graph 3

a) There are 8 edges in the DFS tree of the equation.

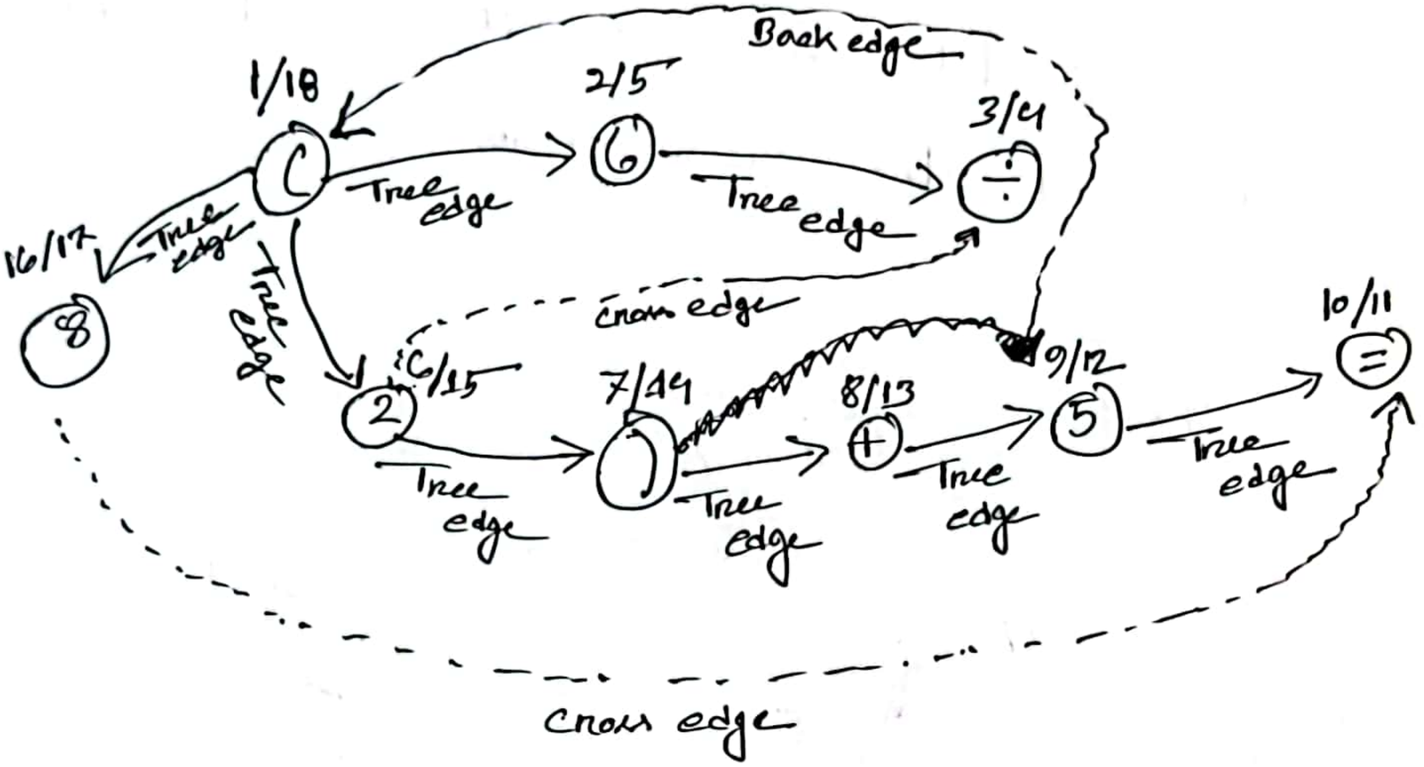
b)



After running a DFS from the equation and making a DFS tree we can say that Bill was right the equation is achievable by running a DFS.

 21201240
Farhad

c) From the DFS Tree we can see that



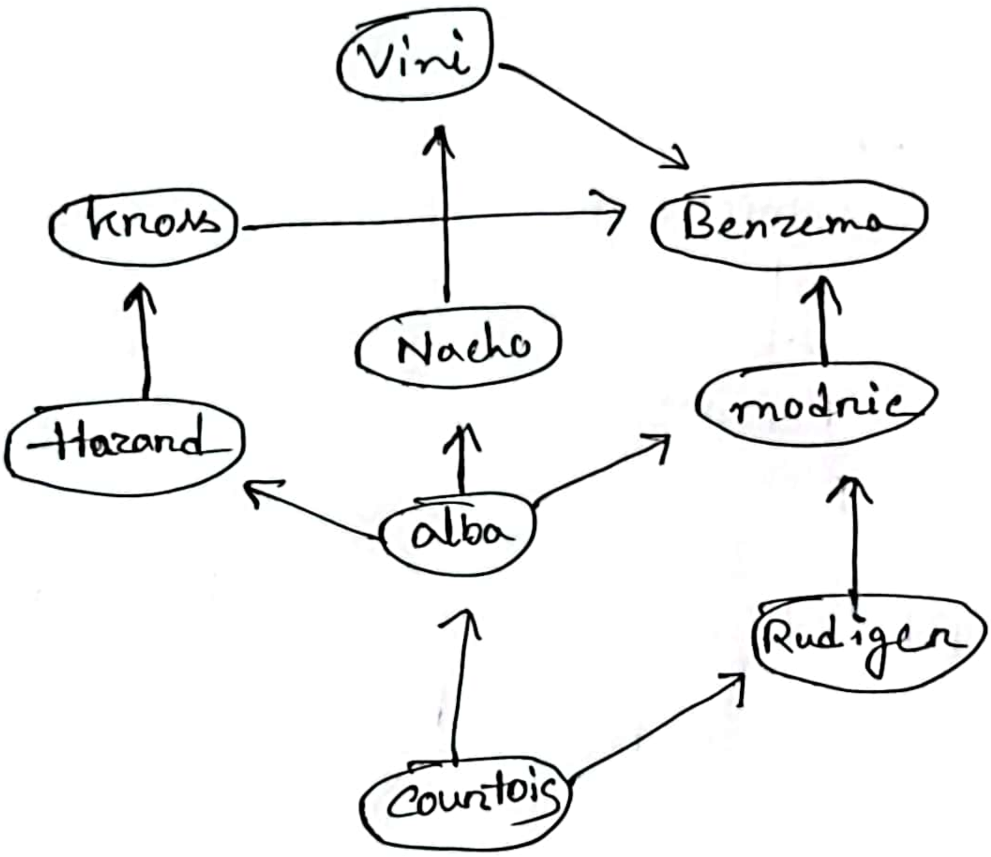
From DFS tree we can see that there are 2 cross ~~and~~ edges from node "8" to node "=" and node "2" to node "÷", and there are one back edge from node "5" to node "C" and rest of all means total 8 tree edges.

⑦ 21201240

Graph 1 :

a) we will use BFS algorithm to know the shortest path.

Here is the BFS Graph



source

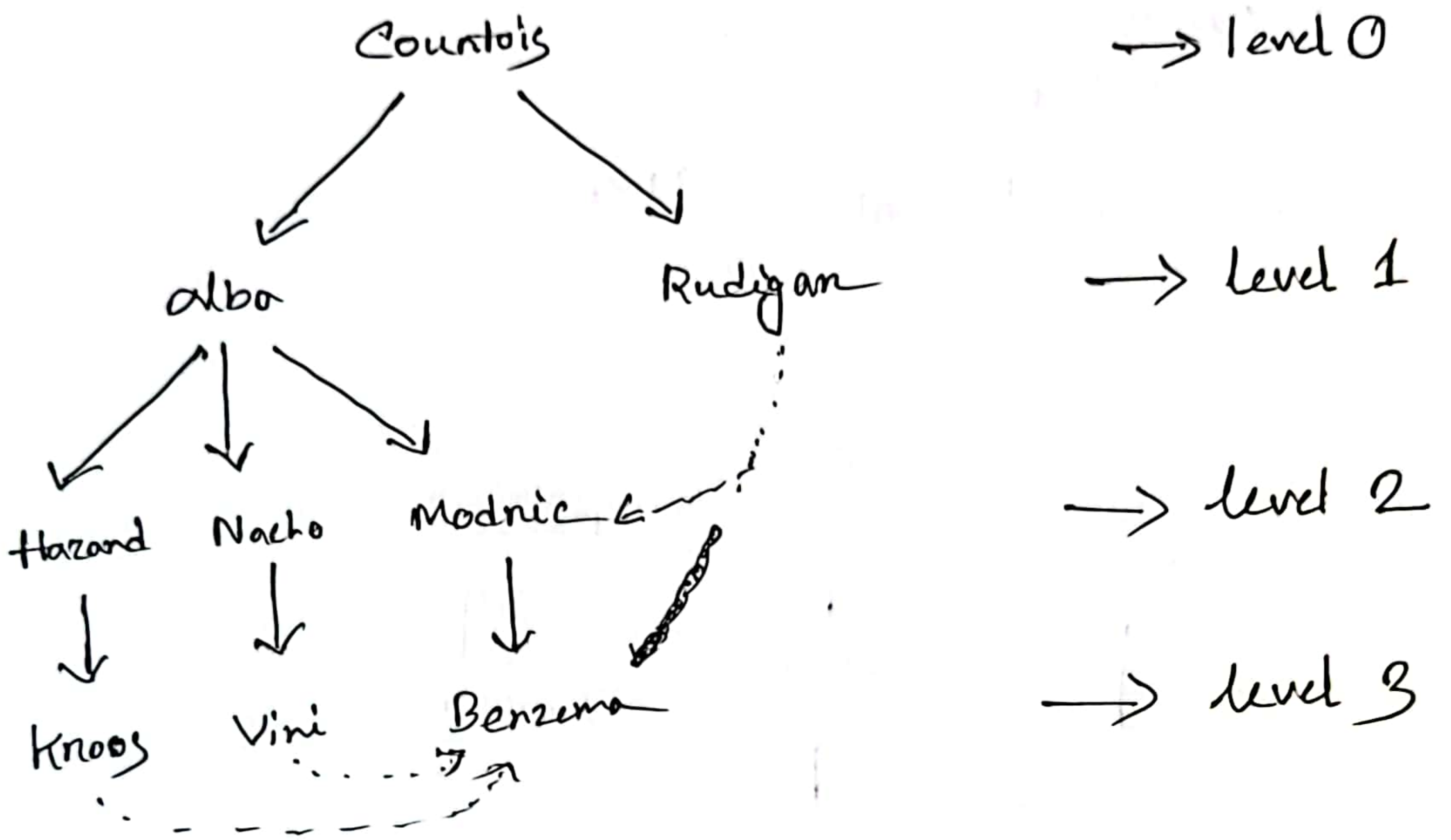
Now draw BFS tree :

Q

~~cortois~~	~~alba~~	~~Rudiger~~	~~Hazard~~	~~Nacho~~	~~Modric~~	~~Kross~~	~~Vini~~	~~Benzema~~	

21201240

BFS Tree :



From the BFS tree we can see that minimum 3 passes need to reach the ball from Countois to Benzema.

b) From the BFS tree we know that there are 3 minimum passes need for reach the ball from Courtois to Benzema. The player who are required for this minimum passes shown below.

Courtois
↓
Rudigar
↓
~~Benzema~~
Modric
↓
Benzema

or

Courtois
↓
alba
↓
Modric
↓
Benzema

21201240

Sorting 2

a) we will do the bubble sort algorithm in this list, as the time complexity of Bubble sort is $O(n^2)$.

steps are shown below.

23	2	19	3	7	11	5	13

↓

2	23	19	3	7	11	5	13

↓

2	19	23	3	7	11	5	13

↓

2	19	3	23	7	11	5	13

↓

2	3	19	23	7	11	5	13

↓

2	3	19	7	23	11	5	13

↓

2	3	19	7	~~23~~ 11	23	5	13

⋮

2	3	5	7	11	13	19	23

b) To sort the given array in linear time, we can use merge function. as we know the time complexity of merge function is O(n)

First we can divide the array into individual subarrays, then we will merge adjacent subarrays pairwise, creating new sub arrays.

[23], [2], [19], [3], [7], [11], [5], [13]

↓

[2, 23], [3, 19], [7, 11], [5, 13]

↓

[2, 3, 19, 23] [5, 7, 11, 13]

↓

[2, 3, 5, 7, 11, 13, 19, 23]

c) We will use binary search to add 15 into the sorted list.

steps are shown below :

1) L m R
[2, 3, 5, 7, 11, 13, 19, 23]

left = 0
Right = 7
mid = 3

7 < 15

21201220

(ii) left = middle + 1 = 4

Right = 7

mid = 5

L m R

[2, 3, 5, 7, 11, 13, 19, 23]

13 < 15

(iii) left = middle + 1 = 6

Right = 7

mid = 6

L/m R

[2, 3, 5, 7, 11, 13, 19, 23]

19 > 15

(iv) Right = mid − 1 = 5

Left = 6

mid = 6

R L/m

[2, 3, 5, 7, 11, 13, 19, 23]

Now insert the value into index 6

[2, 3, 5, 7, 11, 13, 15, 19, 23]

The updated sorted array is [2, 3, 5, 7, 11, 13, 15, 19, 23] with the element 15 inserted at the correct position.

21201240
Tom

Sorting 3

(b) Jimmy's strategy of modifying bubble sort and insertion sort algorithm to find the first five largest and five lowest number in just 5 iteration, this approach is not reliable because:

(i) Bubble and insertion sort are ~~di~~ designed to sort the whole entire array not to finding the largest or smallest elements.

(ii) Only in 5 iteration the sorting will not complete fully. Consequently, the rightmost and left most 5 numbers may not actually represents the largest ~~or~~ and smallest elements respectively.

(iii) The largest elements may not necessarily occupy the rightmost position after a limited ~~m~~ number of iterations and the smallest elements may not occupy the leftmost positions. The order of elements can be arbitrary within the partially sorted subarray.

(iv) After 5 iteration we can see the five right most number will be [12,15,6,10,18] and left most number will be [2,3,4,7,9]

so, as we can see from this, Jimmy's strategy of modifying bubble sort and insertion sort is not ~~not~~ reliable.

(c) (i) In the given List: [13,11,19,7,23,37,29, 53,59,41]

in this list the first element is 13.
for 13 to be placed in its correct position, it should be greater than 11,7 and ~~less~~ smaller than (23,37,29,53 59,41).
∴ The pivot before the first partition was 13.

(d) [13, 11, 19, 7, 23, 37, 29, 53, 59, 41]

13 — Pivot

7, 11

19, 23, 37, 29, 53, 59, 41

After partitioning using 13 as the pivot the list will look like:

[7, 11, 13, 19, 23, 37, 29, 53, 59, 41]

Ans