Assignment - 01

CSE221: Algorithms

Name	: Mohammed Abdul Al Arafat Tanzin
ID	: 21301178
Section	: 03

Date: 10th July 2023

①

ID: 21301178

Time complexity -1

(a)

number of students, $n = 400000$

number of awards. $= 50$

Here the time complexity of most efficient sorting algorithm is $O(n \log n)$.

So, time complexity of most efficient for students

$= O(400{,}000 \log 400000)$.

After that the top 50 students are given in a constant time.

So, ~~the~~ time complexity for awards $= O(50)$

$\therefore$ Total time complexity $= O(400000 \log 400000) + O(50)$

$= O(400000 \log 400000)$

ID : 21301178

(b)

Outer loop iterates len(elements)

Inner loop iterates len(elements)

Hence those two loops are nested

∴ time complexity = len(elements) * len(elements)

$= O(n^2)$

(a)

$T(n) = 625\,T(n/5) + n^3$

Comparing this with $T(n) = aT(n/b) + cn^k$

Here, $a = 625$, $b = 5$; $k = 3$

Since, $b^k < a$

∴ Time complexity $= O(n^{\log_b a}) = O(n^{\log_5 625}) = O(n^{\log_5 5^4})$

$= O(n^4)$

ID:21301178

Time complexity-2

(a)

Here,

For the 1st loop $= O(\log_7 n)$

" " 2nd loop $= O(n/3) = O(n)$

" " 3rd " $= O(n/1) = O(n)$

" " 4th " $= O(n/5) = O(n)$

Here 1st & 2nd & 3rd loops are nested and then the 4th

$\therefore$ Time complexity $= [O(\log_7 n) * O(n)] * O(n) + O(n)$

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

(Ans)

(b)

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

Here,

Using Masters' theorem we get $a=1, b=4, k=1$

and $b^k > a$

$\therefore$ time complexity $= O(n)$

Again applying masters theorem on $T(n/2) + n$

$a=1 ; b=2 ; k=1$

$(2)^1 > 1 \Rightarrow b^k > a$

$\therefore$ time complexity $= O(n)$

$\therefore$ total time complexity $= O(n)$ (Ans)

M1

ID 21301178 ;

Sorting -1

(a)

To order the tracks using selection sort

0	1	2	3	4	5	6	7
15	18	112	33	39	88	8	

0	1	2	3	4	5	6	7
15	18	11	112	33	39	88	41

Pass -1

0	1	2	3	4	5	6	7
8	15	11	112	33	39	88	41

↔ Sorted

Pass -2

0	1	2	3	4	5	6	7
8	11	15	112	33	39	88	41

↔ Sorted

Pass -3

0	1	2	3	4	5	6	7
8	11	15	112	33	39	88	41

↔ Sorted

Pass -4

0	1	2	3	4	5	6	7
8	11	15	33	112	39	88	41

↔ Sorted

Pass : 5

8	11	15	33	39	112	88	41

Sorted

Pass -6

8	11	15	33	39	41	88	112

Sorted

Pass -7

8	11	15	33	39	41	88	112

Sorted

(b)

From (a), we can see loop execute in following order:

$$1+2+3+\dots+(n-1)$$

$$\Rightarrow \frac{n(n-1)}{2}$$

$$\Rightarrow \frac{n^2-n}{2}$$

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

(V) (IV)

ID: 21301178 ;

Sorting-2

(a)

Here selection sorting is suitable

(b)

0	1	2	3	4	5	6	7
23	2	19	3	7	11	5	13

Pass-1

2	23	19	3	7	11	5	13

Sorted

S-2

2	3	19	23	7	11	5	13

Sorted

S-3

2	3	5	23	7	11	19	13

Sorted

S-4

2	3	5	7	23	11	19	13

Sorted

S-5

2	3	5	7	11	23	19	13

Sorted

Pam-6

$$\underbrace{\begin{array}{|c|c|c|c|c|c|c|c|}\hline 2 & 3 & 5 & 7 & 11 & 13 & 19 & 23 \\ \hline\end{array}}_{\text{Sorted}}$$

(b)

Jack sorted the list in linear time using an algorithm called counting sort. Here are the steps

S-1| Find the maximum and minimum values in the list : max = 23 and min = 2

S-2| Create an array with a length equal to the range of values in the list (length = max-min+1 = 22)

S-3| For each even index i starting from 0 up to n-2 (eg : i = 0, 2, 4, ...)

S-4| For each odd index i starting from 1 up to n-2 (e.g : i = 1, 3, 5, 7 .....)

ⓥ : 21301128

5-5) For each index i starting from 0 up to array-1 (eg : i = 0, 1, 2, ... ,21)

5-6) The output list is finally sorted

(c)

From (1) we got sorted list. [2, 3, 5, 7, 11, 13, 19, 23]

If Jack wants to add 15 into this sorted list, then he needs to follow Quick Sort algorithm.

~~B~~ By using Quick Sort,

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

0	1	2	3	4	5	6	7	8
2	3	5	7	11	13	15	19	23

∴ At 6th index, 15 will be sorted

ID: 21301178

Sorting-3

(b) I support the strategy, because by applying bubble sort to a array everytime ~~of~~ the largest number of that array get positioned respectively. ~~of~~ To find the 5 largest or smallest number we have ~~apply~~ to iterate only 5 times the 5 ~~to~~ most largest number of the array will get their position.

(c) Before the first partition 23 was the pivot of the given array? We know that pivot get sorted position after ~~every~~ the partition function is called. We know 23 got it's posititon as every left element of 23 is smaller than 23 and ~~aig~~ all the elements of the rightside is ~~to~~ greater than 23.

(d) After calling the partition function again using 13 as pivot the resultant array will look like this

~~11 7 19~~

7	11	13	19	23	37	29	53	59	41

13 will get it's sorted position.

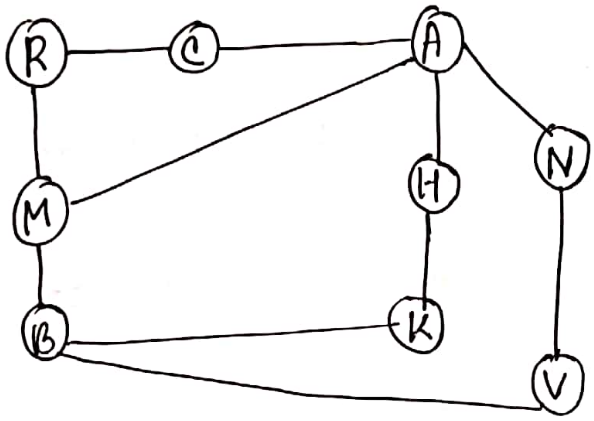
— o —

Graph - 1

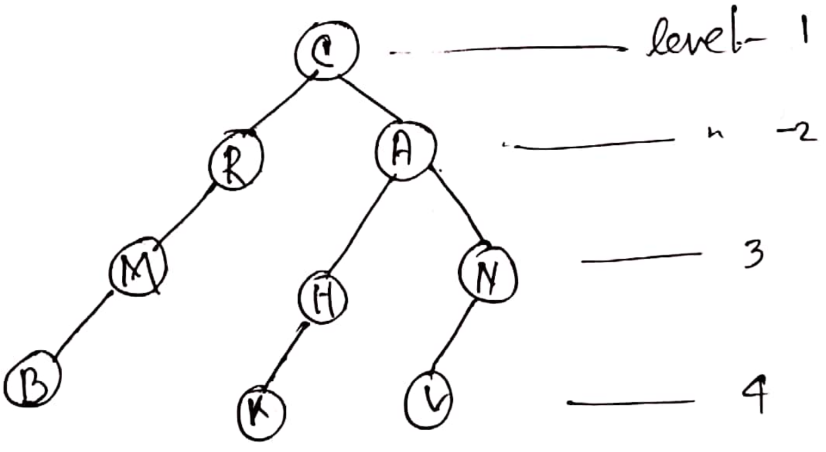
ID: 21301178 ;

(a)

By using BFS, we can draw the graph. Here All players are denoted by the first letter of their names,



~~V~~
~~K~~
~~B~~
~~N~~
~~H~~
~~M~~
~~A~~
~~R~~
~~C~~



(b)

Player are required for minimum number of pases are

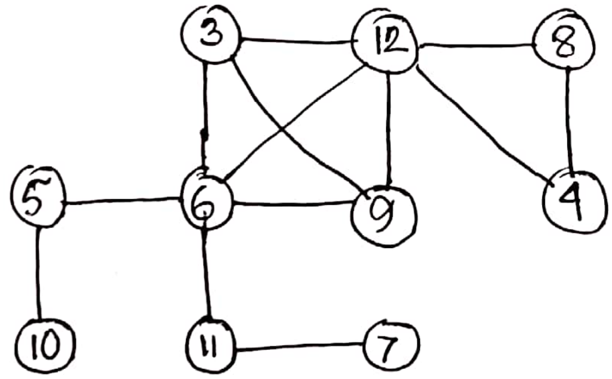
Courtois → Rudiger → Modric → Benzema

Total 4 players are required. ~~[illegible]~~

-o-

Graph-2

(a)



(b)

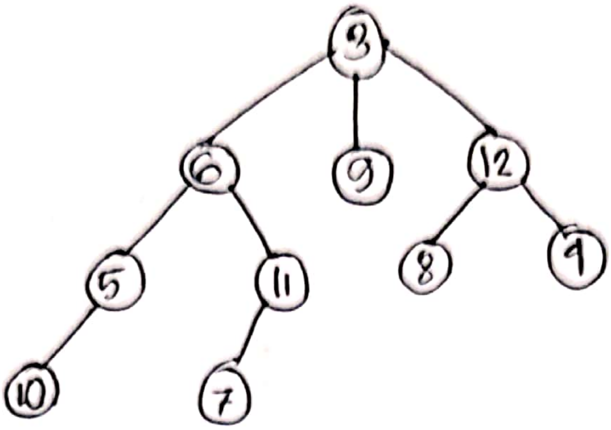
Yes Bill right. There are at least 4 triangles. these triangles are made of these following nodes:

Triangle 1 (4,8,12)
Triangle2(12,3,6)
triangle(3,6,9)
Triangle(6,9,12)

(c)

ID:21301178 ;

7
10
4
8
11
5
12
9
6
3



Shortest distance from node 3 to

node 6 = 1

node 5 = 2

node 10 = 3

node 11 = 2

node 7 = 3

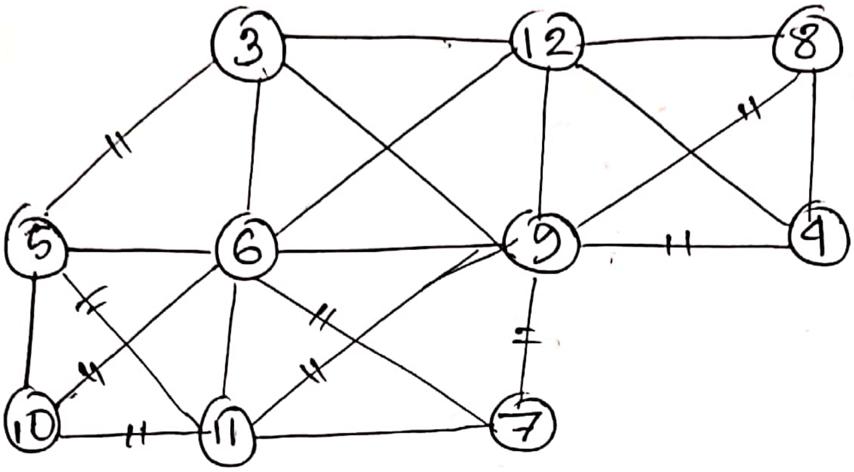
node 9 = 1

node 12 = 1

node 8 = 2

node 4 = 2

(d)



total number of newly added edges = 9

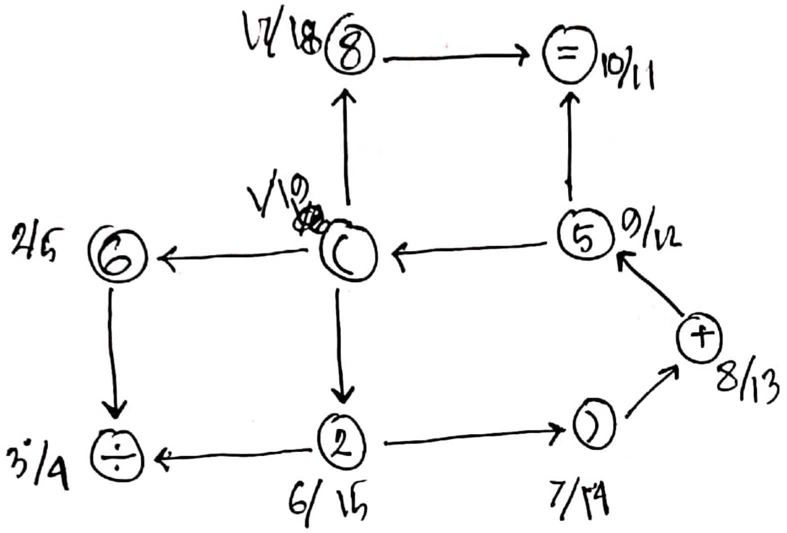
~~Total numbers~~ newly edges are between

(3,5) , (5,11), (6,10), (10,11), (9,11), (6,7), (9,7), (4,9), (8,9)

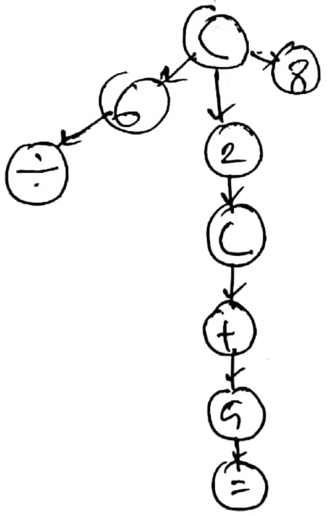
ID: 21301178

Graph-3

(a)



DFS tree



The DFS tree contains 8 edges

(b)

Yes, Bill is right. The equation is achieavable by running DFS.

validation Starting from the source node '(' and from the source node, went '6' and then '÷' now we reached the end, then backtrack to node '(' and then went to node '2' and then ')' next nod '+' and then node '5' and then node '=' ~~B~~ As we reached the end we again backtrack to nod '(' ~~&~~ then we reached node '8'

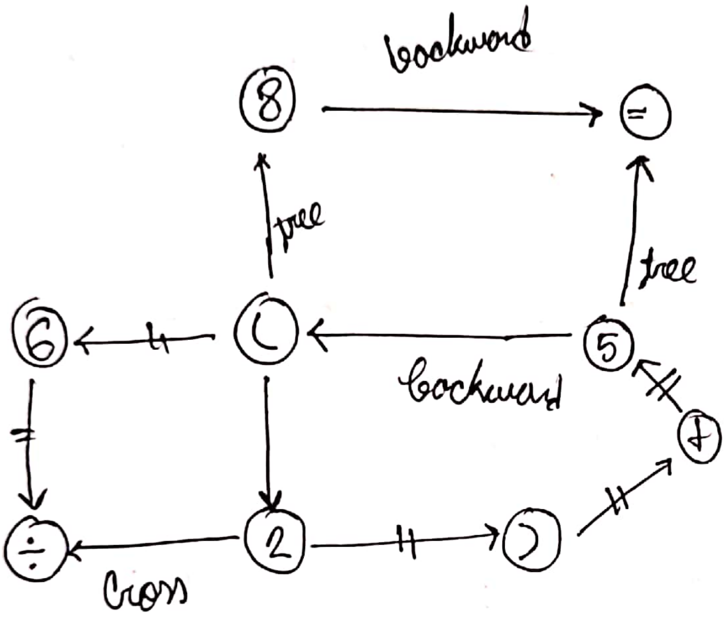
Order of the edges :

(,6 → 6, ÷ → (, 2 → 2,) →), + → +, 5 → 5, = → (,8

ID: 21301178 ;

(c)

From (a), we got the DFS tree. Now, here is the edge classification of the main graph:



From the classified graph, we can see that there are 8 Tree Edges, 2 backward Edges, 1 cross Edge but forward edge.