ARTIFICIAL INTELLIGENCE
ASSIGNMENT 02

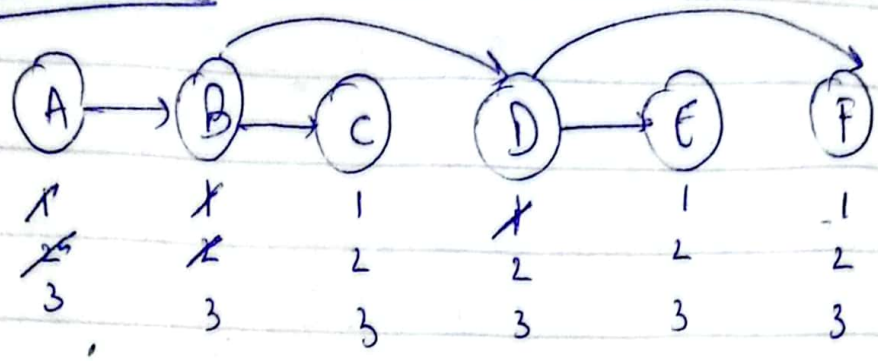
BILAL AHMED KHAN (20K-0183; Sec: B)

QUESTION NO. 01

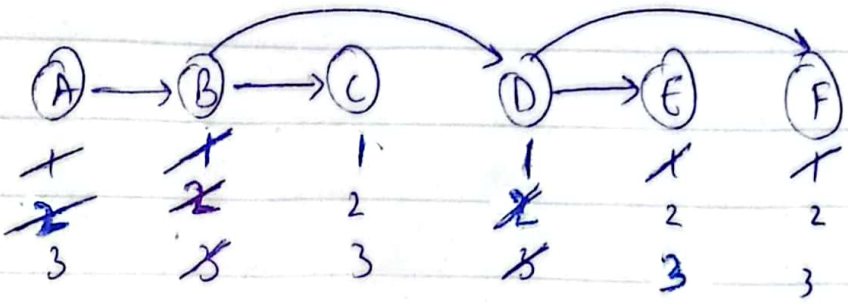
When checking arc consistencies from leaves, it reduces the domain in a top-to-bottom manner, all assignments are done by the time root node is reached. Because previous assignments have already been made consistent.

If we start by root assignments then we will have more failures more checks also, we will not know about the failure till we reach the ~~root~~ leave.

from leaves:



from root



Fails at C, and we didn't even know till we reached that stage, thus we always start from leaves.

QUESTION NO. 02

```
 CRACK
  HACK
 ERROR
```

C 0 1 2 3 4 5 6 7 8 9
R 0 1 2 3 4 5 6 7 8 9
A 0 1 2 3 4 5 6 7 8 9
O 0 1 2 3 4 5 6 7 8 9
K 0 1 2 3 4 5 6 7 8 9
H 0 1 2 3 4 5 6 7 8 9
E 0 1 2 3 4 5 6 7 8 9

1) C cannot be 0 and C is being converted into E because of carry thus we'll assume

$$C = 1$$

$$E = C + 1$$

$$E = 2$$

2) Now $R + H + 1 = R$

For $R = 0$

$$0 + H + 1 \neq R$$

For $R = 3$ (since 2 is already taken)

$H = 0$ $\quad 3 + H + 1 \neq 3$

⋮

$H = 9$ $\quad 3 + 9 + 1 = 13$

possible!

$R_2 3$

$H_2 9$

C	R	A	O	K	H	E
1	3				9	2

3) $1+A+A_2 R$

$A_2 0 \Rightarrow 0+0+1 \neq 3$

⋮

$A_2 6 \Rightarrow 6+6+1 = 13$ → i.e. R

$A_2 6$

C	R	A	O	K	H	E
1	3	6			9	2

4) $1+C+C_2 0$

$C_2 0 \quad 1+0+0 \neq 0$ (not parible)

→ Backtrat to $C+1 = E$ (first constraint)

C = 2

E = 2+1 = 3

C	E
2	3

→ R = 0

0+1+H ≠ 0

R = 1

H = 0 ⟶ 1+1+0 ≠ 1

⋮

H = 9 ⟶ 9+1+1 = 11

C	E	R	H
2	3	1	9

→ 1+A+A = R

A = 0 ⟶ 0+0+1 ≠ 1

⋮

A = 5 ⟶ 5+5+1 = 11

C	E	R	H	A
2	3	1	9	5

$C + C_2O$

$2+2 \neq 0$

Not possible, backtracking to c

→ $C = 3$

$E = 1+C = 4$

→ R_2O not possible

R_1

$H_2O \Rightarrow 1+1+0 \neq 1$

$H_2 9 \Rightarrow 1+1+9 = 11$

C	E	R	H
3	4	1	9

→ $A+A+1 = R$

~~A=0~~ A_2O

$0+0+1 \neq R$

⋮

$A_5 \Rightarrow 5+5+1 = 11$

C	E	R	H	A
3	4	1	9	5

C+C = 0

3+3 ≠ 0

(Not possible
backtracking to C)

→ Now C = 4

E = C+1 = 5

R = 0 (not possible)

R = 1 (not possible K+K ≠ 1)

R = 2

H = 0 1+2+0 ≠ 12

⋮

H = 9 1+9+2 = 12

C	E	R	H
4	5	2	9

→ A+A = 1R

A = 6 6+6 = 12

A = 6

C	E	R	H	A
4	5	2	9	6

→ C + C = O

4 + 4 = 8 , 0

O = 8

C	E	R	H	A	O
4	5	2	9	6	8

→ K + K = R

K = 0 (not possible)

K = 1 ⇒ 1 + 1 = 2

(K, 1)

C	E	R	H	A	O	K
4	5	2	9	6	8	1

C	R	A	C	K
+	H	A	C	K
E	R	R	O	R

	1	1		
4	2	6	4	1
+	9	6	4	1
5	2	2	8	2

Proved!!!

QUESTION NO. 03

We will need 25 variables to represent the current problem

The attribute of every variable will be shown such that,
"House-3-color" will represent the color of house 3.
Thus each of the 5 houses will be represent each property i.e. Pet, drink, nationality, color & candy we will get 25 variables

Domains:-

let X be the house no. 1 to 5

House X Color = {red, yellow, ivory, green, blue}

House X Nationality = {Englishman, Ukrainian, Spaniard, Norweign, Japanese}

House X Candy = {Kit Kat, Hershey, Smarties, Snickers, Milky Way}

House x Drink: { Orange Juice, Tea, Coffee, Milk, Water }

House x Pet = { Horse, Zebra, Dog, Fox, Snail }

Constraints:

We will use binary & unary constraints

- House 3 Drink = "milk"
- House 3 Drink ≠ House 2 Drink
- House 1 Nationality = "Norwegian"
- House x Color = Ivory → House(x+1) Color = green
  For x = 1 to 4
- House x Candy = Hershy → (House(x+1) Pet = Fox OR House(x-1) Pet = Fox)
  For x = 2 to 4
- House x Candy = "KitKat" → (House x Color = "yellow" OR (House(x-1) Pet = Horse OR House(x+1) Pet = House))
  For x = 2 to 4
- House x Nationality = Englishman → House x Color = Red
- House x Nationality = Spaniard → House x Pet = Dog
- House x Nationality = Norwegian → House(x-1) Color = Blue

- House X Candy = Snarlies → House X Pet = "Snail"
- House X Candy = Snickees → House X Drink = "Orange Juice"
- House X Nationality = "Ukraine" → House X Drink = "Tea"
- House X Nationality = "Japanese" → House X Candy = "Milky Way"
- House X Color = "Green" → House X Drink = "Coffee"

This means that 5^{25} possible assignment & $O(25^2)$, 625 possible constraints which is very inefficient.

2nd approach:

Instead we can have 5 variables for each house

House X = {Color, Nationality, Candy, Pet, Drink}

where X = 1 to 5

→ 5 variables with 5 domain sizes = 5^{15} assignment & $5^2 = 25$ max possible constraints

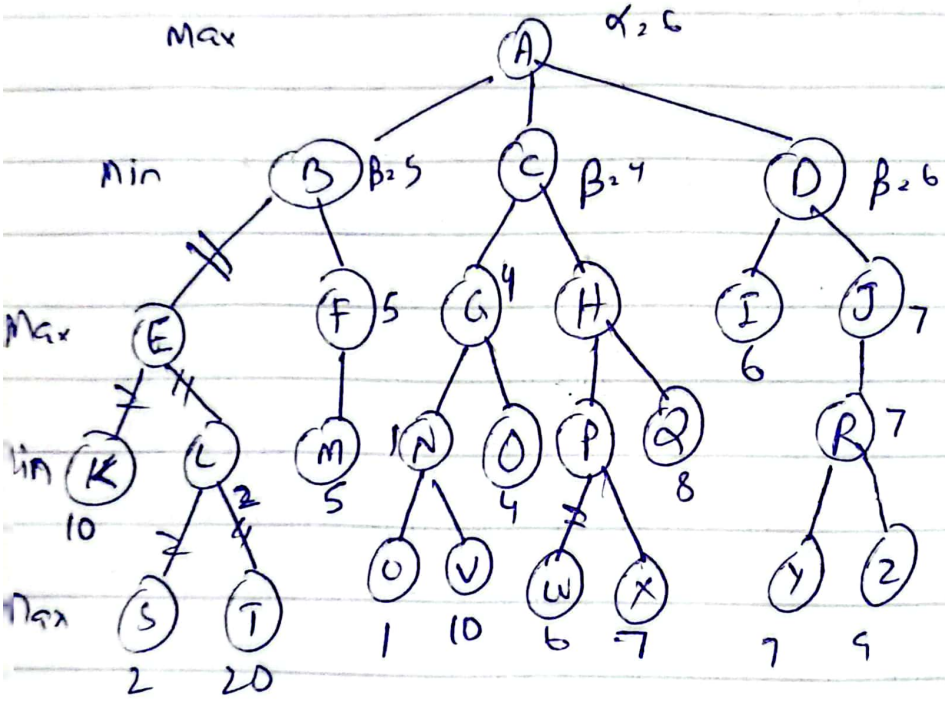
Now,

Search space	25	v/s	5 variable
Constraints	25	v/s	625
Time complexity	5^5	v/s	5^{25}

Thus we should go with the "2nd approach"

QUESTION NO. 04

a) 6
b) A → D
c) D → I
d) E, K, L, S, T, W



Q5

a) Resource allocation btw different kind of users i.e. licensed & cognitive users & how licensed user utilize different channels through different game theory techiques.
Different challenges faced when apply game theory to a dynamically allocated resources of a network are also discussed. Thus also sheds light on the acess of liscensed users & maximizing cognitive users QoS.

b) To perform communication btw CRN nodes Non-cooperative game theory is used. each node tries to maximise their evaluation function with their attributes being power level, follow controlek.
The nodes dont have access to each other's strategies. The SINR represents the efficiency of the spectrum to reach Nash equilibrium state.

c) Underlay allows CRs & WS to coexist & is tolerant of CR interference since it focuses more on WU protection & CR at the same time.

On the other hand, Overlay gives priority to WU & only allows allows CR to access unoccupied bands left by WU.

QUESTION NO. 06

a)

The first paper discusses how player's performance & time complexity can be increased using alpha beta pruning.
While the second paper discusses the parallelization of alpha beta pruning & how using multiple computers affect its efficiency in multiplayer games

b) Sequential approach tries to solve the complete tree on a single thread thus its slower, while openMP allocates a part of tree to each thread thus multiple threads are solving the tree at once which is quicker in comparision

c) Sequential approach assigns the whole tree to a single thread therefore its slower, CUDA allocates different parts of the tree to GPU which traverses the tree using CUDA cores thus its faster.