CSE 422 ASSIGNMENT 1

NAME: ANIKA ISLAM
ID: 21101298
SECTION: 06

Question 1

(1)

	A	B	C	D	E	F	G	H
C1 → ABH	1	1	0	0	0	0	0	1
C2 → ABCDH	1	1	1	1	0	0	0	1
C3 → EFH	0	0	0	0	1	1	0	1
C4 → DEGH	0	0	0	1	1	0	1	1

(2) $We = \sum_{i=1}^{8} W(O_i) * B(O_i)$

$max\_weight = 12$

$Re = \sum_{i=1}^{8} Re(O_i) * B(O_i)$

$$fitness, f(n) = \begin{cases} 0 - (We - max\_weight), & We > max\_weight \\ Re, & We \leq max\_weight \end{cases}$$

	We	We > max-weight	We ≤ max-weight	f(n)
C1	5	False	True	32
C2	16	True	False	-4
C3	13	True	False	-1
C4	32	True	False	-10

C4 has the least fitness function, so is excluded.
C1 and C3 have the best fitness function

(3) C1 × C3 → A B E F H H → AFH EBH

O1 → AFH

O2 → EBH

(4) O1 → AFH → ADH (2nd gene is mutated)

Randomly, a gene is selected from both offsprings and that gene is changed due to mutation.

O2 → EBH → EAH (2nd gene is mutated)

mut − O1 → ADH

mut − O2 → EAH

	We	We > max-weight	We ≤ max-weight	f(n)
mut - O1	11	False	True	67
mut − O2	10	False	True	42

On checking the fitness function of both the mutated offspring, mut-O1 have the highest fitness function. During crossover, mut-O1 have received the best genes from parents and during mutation, the change in gene caused an advantage to be the best in this scenario.

∴ best offspring = mut - O1 → ADH

Question 2

A* Graph Search

visited :- $A^7 | E^6 | H^6 | C^9 | F^8 | G^9 | X^8$

A^7

A^7 B^{10} C^9 E^6

E^6 B^{10} C^9 F^9 H^6

H^6 B^{10} C^9 F^9 D^{12} X^9

C^9 B^{10} ~~A^7~~ D^{12} X^9 F^8 G^9

F^8 B^{10} D^{12} X^9 G^9

G^9 B^{10} D^{12} ~~X^9~~ X^8

X^8 B^{10} D^{12}

path : A ⟶ C ⟶ G ⟶ X

cost : 8

Greedy Best first search

visited:- $A^7 | E^4 | H^2 | X^0$

A^7

A^7 B^5 C^6 E^4

E^4 B^5 C^6 F^4 H^2

H^2 B^5 C^6 F^4 D^3 X^0

X^0 B^5 C^6 F^4 D^3

path: A → E → H → X

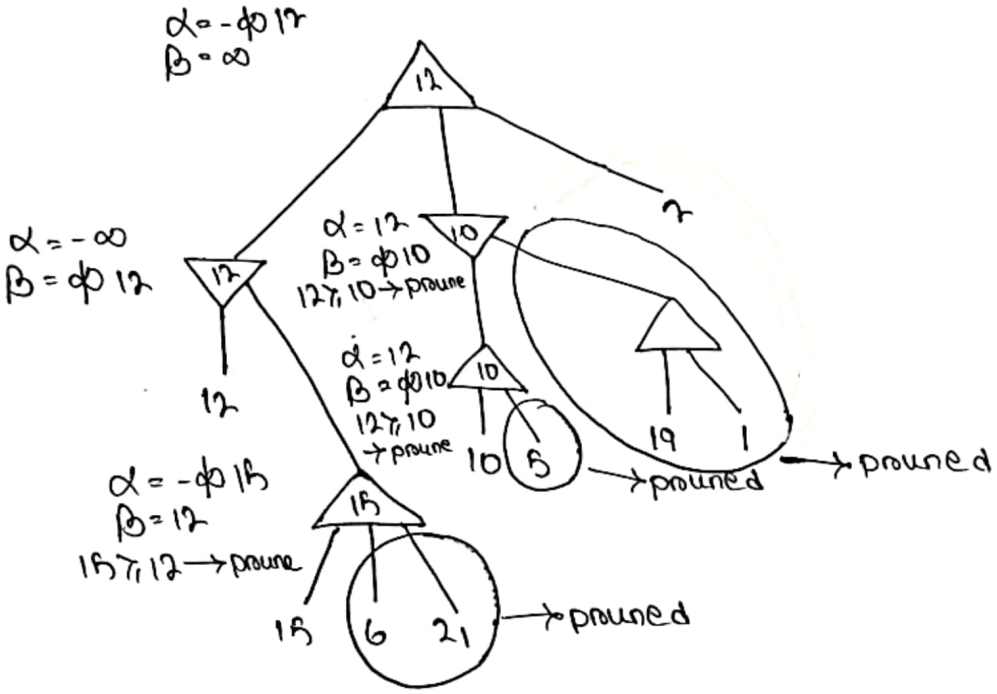
cost: 9

Question 3

$\alpha = -\infty$ [update with max value at Δ]

$\beta = \infty$ [update with min value at ∇]

$\alpha \geq \beta \rightarrow$ prune



Question 4

(1)

C1 ⟶ 1111

C2 ⟶ 0010

C3 ⟶ 0011

C4 ⟶ 1000

(2) bin_to_dec = int(0') ; 0 → chromosome

x = bin_to_dec

$$\text{fitness}_1 = g(n) = \begin{cases} -x, & f(x) \neq 0,\ f(x) \in \mathbb{R} \\ x, & f(x) = 0 \end{cases}$$

	$f(x)$	fitness, $g(n)$
C1	156	-15
C2	0	2
C3	0	3
C4	30	-8

C1 has the lowest fitness function, so is excluded.

C2 and C3 have the best fitness function.

C2 × C3 ⟶ 0 0 / 0 0 (1 ↔ 1) ⟶ 0011 / 0010

O1 ⟶ 0011

O2 ⟶ 0010

(3) 01 ⟶ 0011 ⟶ 0010

4th gene is mutated

02 ⟶ 0010 ⟶ 0000

3rd gene is mutated

Randomly, a gene is selected from both offsprings and that gene is changed due to mutation.

mut – 01 ⟶ 0010

mut – 02 ⟶ 0000

	f(x)	fitness, (f(x))
mut – 01	0	2
mut – 02	6	0

On checking the fitness function of both the mutated offspring, mut – 01 have the highest function.

During crossover, mut – 01 have received the best genes from parents and during mutation, the change in gene caused an an advantage to be the best in this function.

∴ best offspring = mut – 01 ⟶ 0010

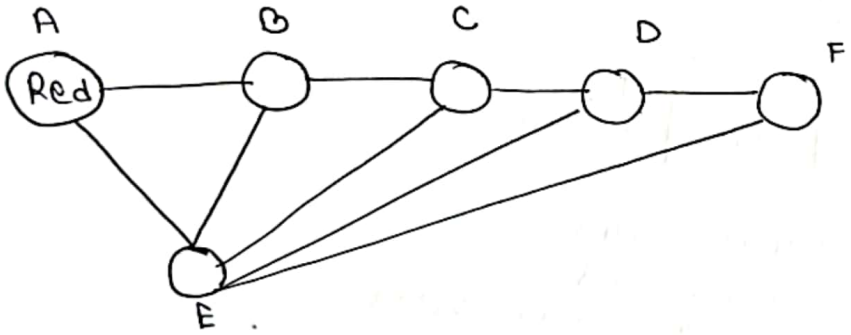
Question 5.

Forward checking can reduce the domain of unassigned variable using assigned variable.

We can basically predict the outcome of the next unassigned variable by using a known variable beforehand.

Node consistency checking is the forward checker where a random node is chosen and color is assigned to it and its following nodes are reduced with respect to the color assigned, and constraint,

Constraint: Two adjacent nodes cannot be of same color



A	B	C	D	E	F
Red	~~RGB~~	RGB	RGB	~~RGB~~	RGB

↓ C is randomly choosen & assigned with green color

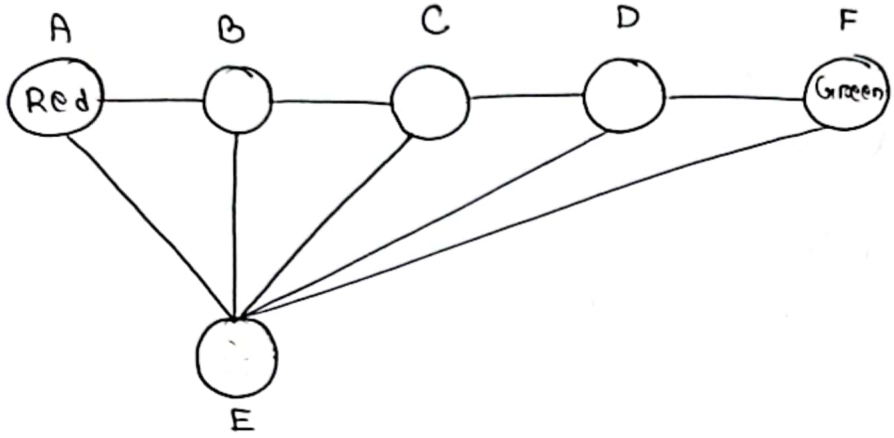
A	B	C	D	E	F
Red	~~RGB~~	Green	~~RGB~~	~~GB~~	RGB

↓ E is randomly choosen & assigned with blue

A	B	C	D	E	F
Red	~~B~~	Green	~~R~~	blue	~~RGB~~

→ B becomes empty; algo stops and graph is node - inconsistent.

Arc consistency Node is stronger than Node consistency checking where domain is checked and remaining value can be used to reduce the domain.



A	B	C	D	E	F
Red	~~RGB~~	RGB	R~~GB~~	~~RGB~~	Green

⇓

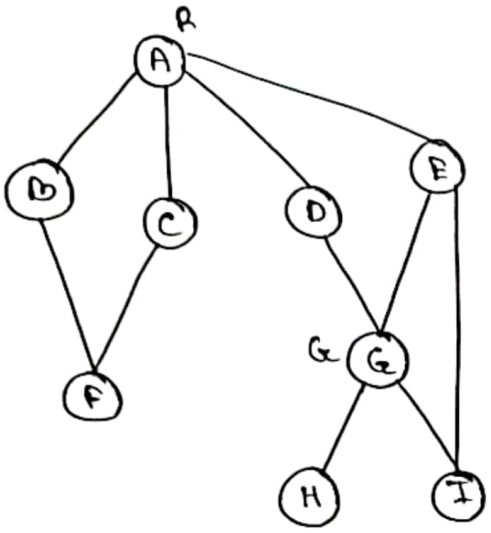
A	B	C	D	E	F
Red	G~~B~~	RG~~B~~	R~~B~~	Blue	Green

⇓

A	B	C	D	E	F
Red	Green	~~RG~~	Red	Blue	Green

→ C becomes empty, algo stops and graph is arc-inconsistent

(1)



Degree heuristic is the most constraining variable. It is the variable which can reduce the highest number of variables.

→ A can reduce 4 variables

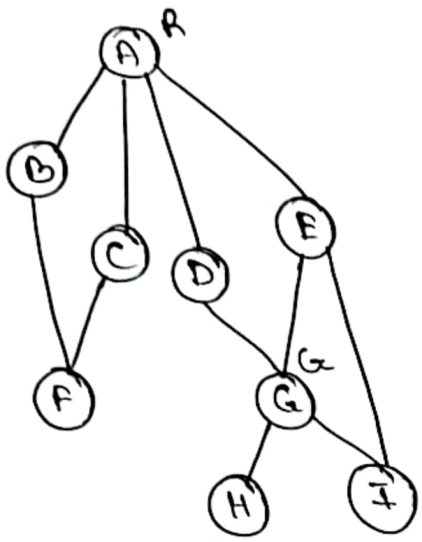
(4) A Red ↑ assigned node	(1) B ~~RGB~~	(1) C ~~RGB~~	(0) D ~~RGB~~	(1) E ~~RGB~~	(2) F RGB	(4) G Green ↑ assigned	(0) H RGB	(1) I RGB

(4) A Red	(1) B GB	(1) C GB	(0) D ~~GB~~	(1) E ~~GB~~	(2) F RGB	(4) G Green ↑ node	(0) H ~~RGB~~	(1) I ~~RGB~~

(4) A Red	(0) B GB	(0) C GB	(0) D B	(1) E B	(2) F Red ↑ assigned	(4) G Green ↑ node	(0) H RB	(1) I RB

(4) A Red	(0) B GB	(0) C GB	(0) D B	(1) E Blue ↑ assigned ↑ node	(2) F Red ↑ node	(4) G Green	(0) H RB	(0) I ~~RB~~

(2)



The color of node "B" can either be green or blue.

If node "B" is assigned with either green or blue, then B can reduce only node "f". So, either blue or green can be assigned to node B.

Here, node B is assigned with blue randomly.