4. $A \to F = f(n) = 13 + 0 = 13$

$A \to B \to D \to E = f(n) = 1 + 2 + 5 + 6 = 14$

$A \to B \to C \to E = f(n) = 1 + 1 + 3 + 6 = \underline{11}$

5. $A \to F = f(n) = 13 + 0 = 13$

$A \to B \to D \to E = f(n) = 1 + 2 + 5 + 6 = 14.$

$A \to B \to C \to E \to F = f(n) = 1 + 1 + 3 + 2 + 0 = \underline{7}.$

:

1) $x(n) - x_i$ np. Max(n) − np. min(x)

2) $$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$$

np. var(n)

3) $S = \sqrt{\sigma^2}$

4) $Q3 - Q1$

5) $$CV = \frac{S.D}{Mean} \times 100\%$$

A* Search:

> In A* search Algorithm, we use search heuristic h(n) as well as the cost to reach the node g(n). Hence we can combine both costs as following, and this sum is called as a fitness number f(n).

> It has combined features of uniform cost search and greedy best-first search.

$$f(n) = g(n) + h(n).$$

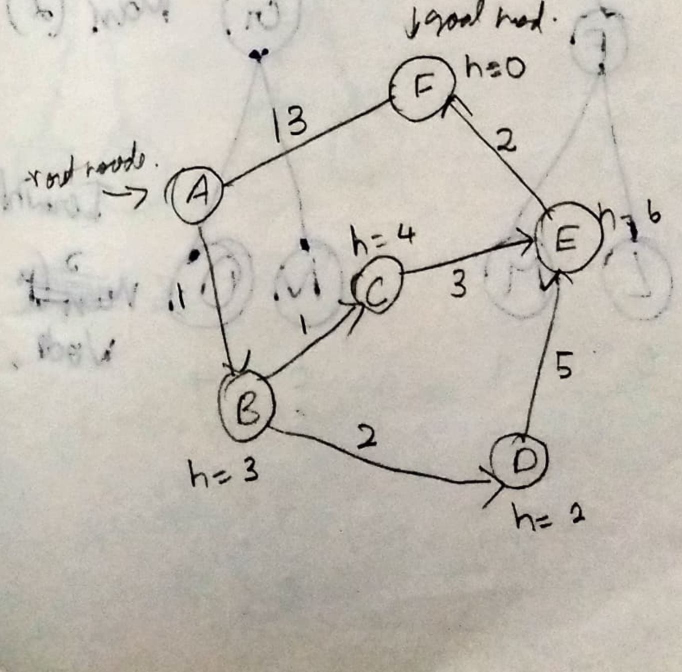
where,

g(n) → cost of the path from start node to node n.

h(n) → cost of the path from node n to goal state (heuristic function.

Eg

$F(n) = g(n) + h(n).$

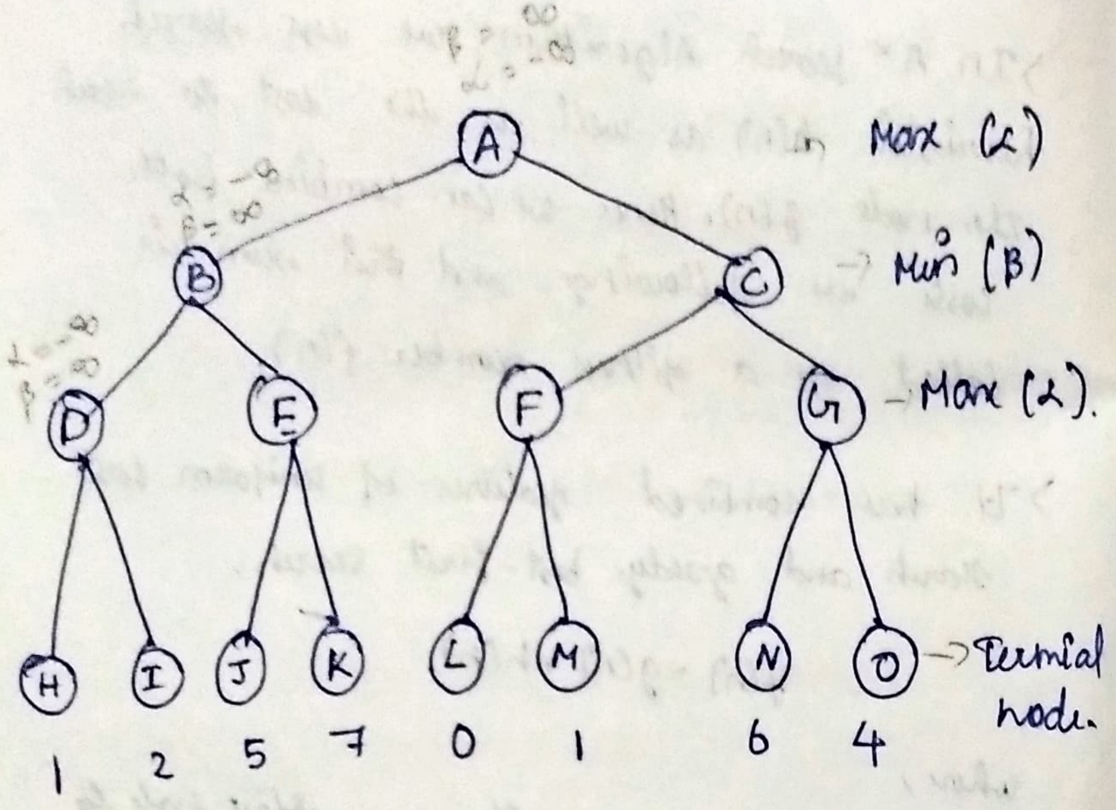


1) A→B = f(n) = 1+3 = 4
A→F = f(n) = 13+0 = 13.

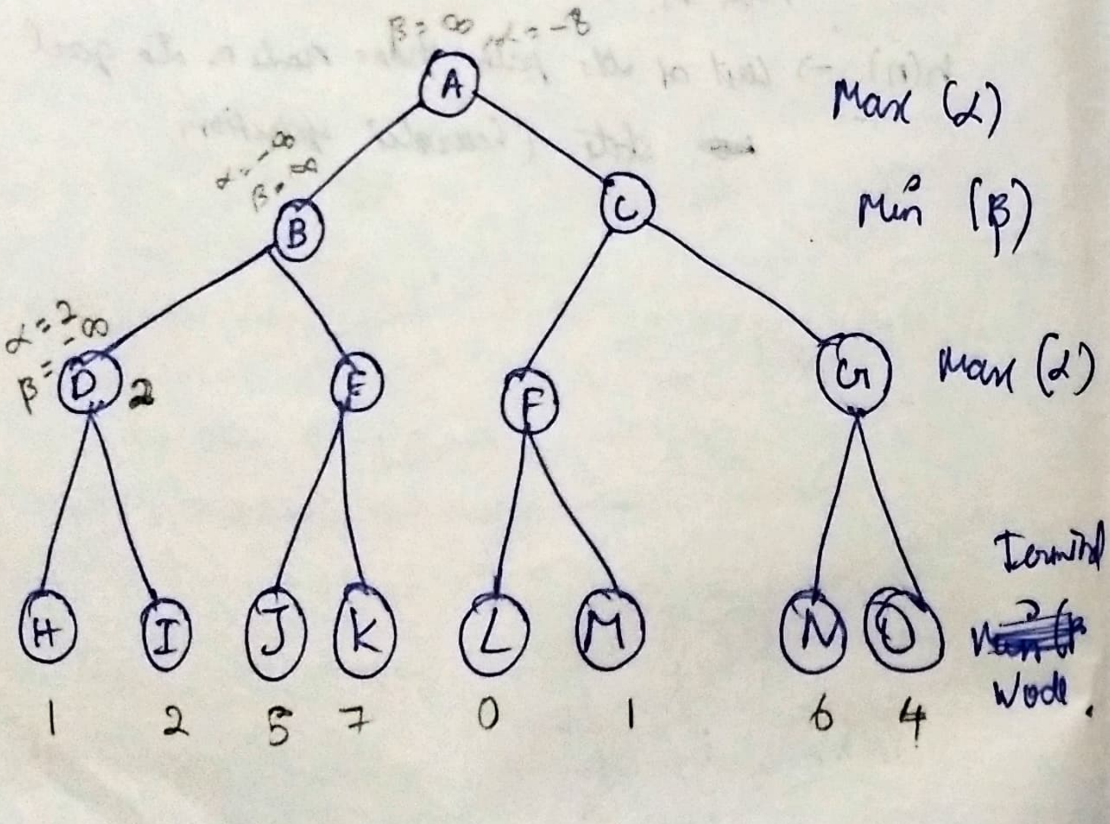
2) A→F = f(n) = 13.
A→B→C = 1+1+4 = 6
A→B→D = 1+2+2 = 5

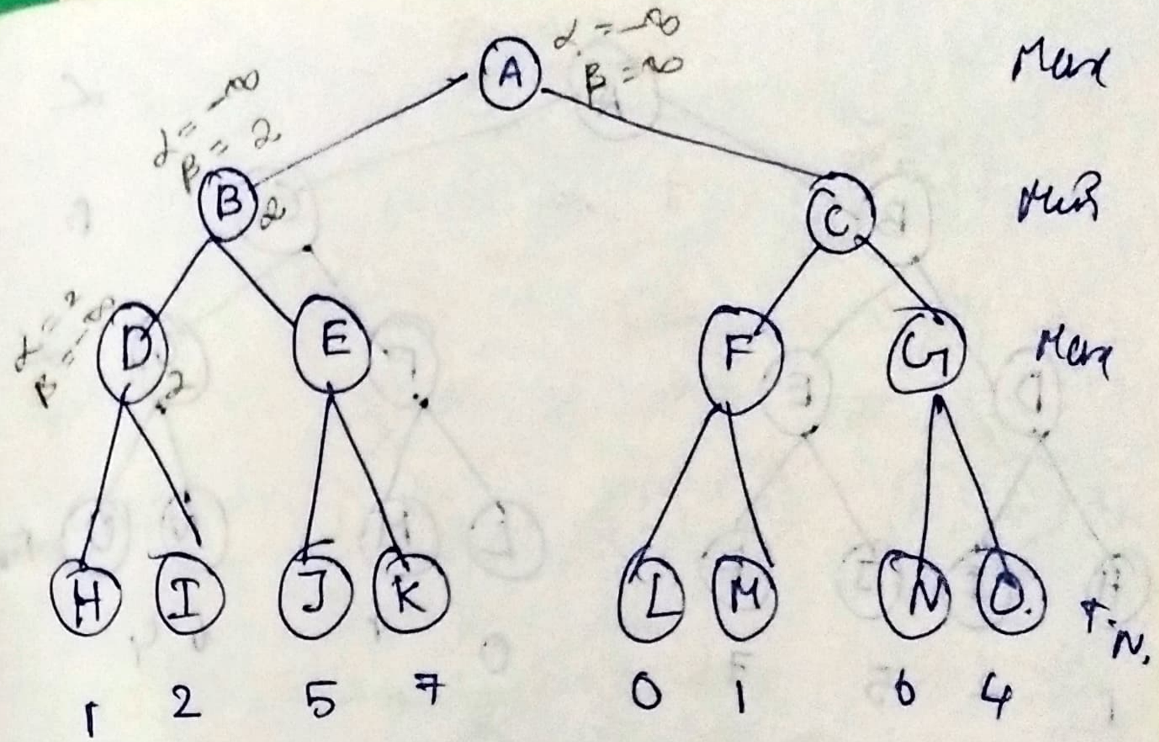
3) A-F = f(n) = 13+0 = 13
A→B→C = f(n) = 1+1+4 = 6.
A→B→D→E = 8+6 = 14

Alpha Beta Pruning:

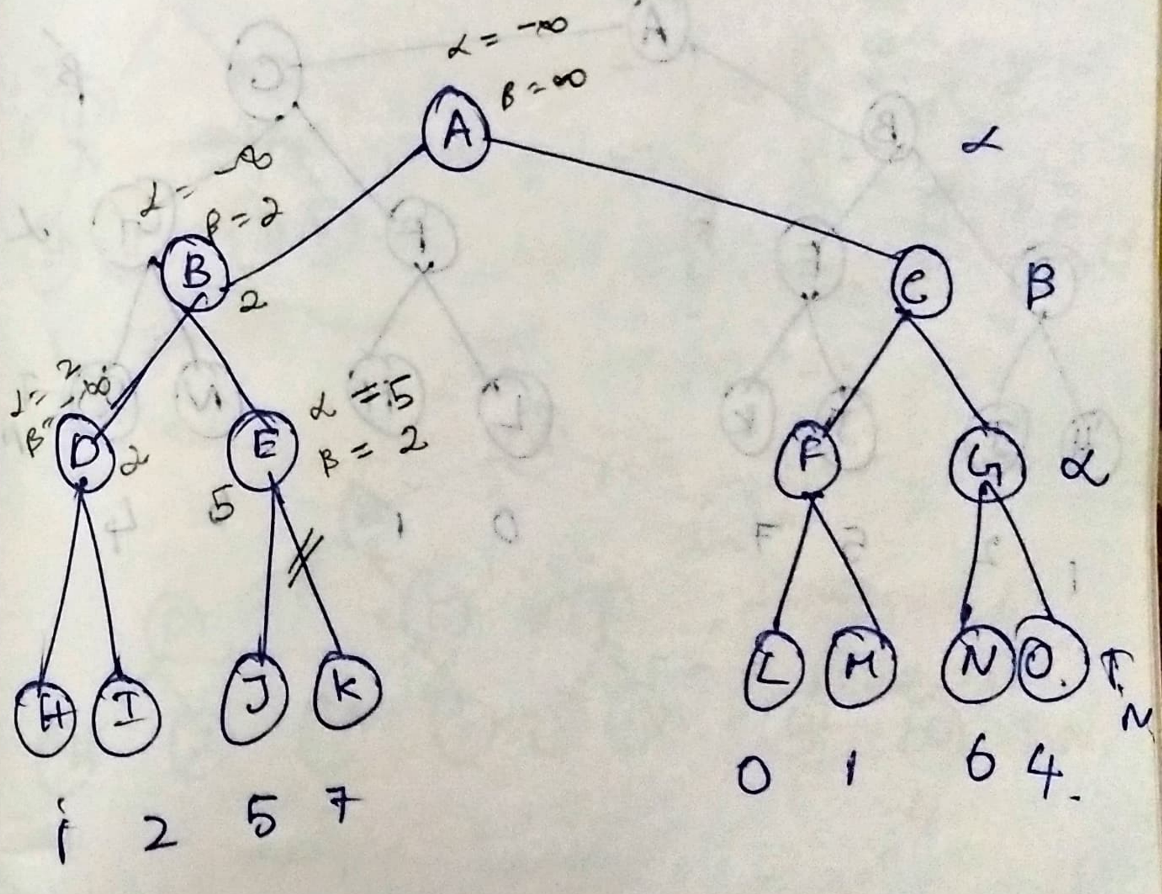


$(-\infty, 1) = \alpha = 1$, $(1,2) = \alpha = 2$, $\beta = \infty$ (1)





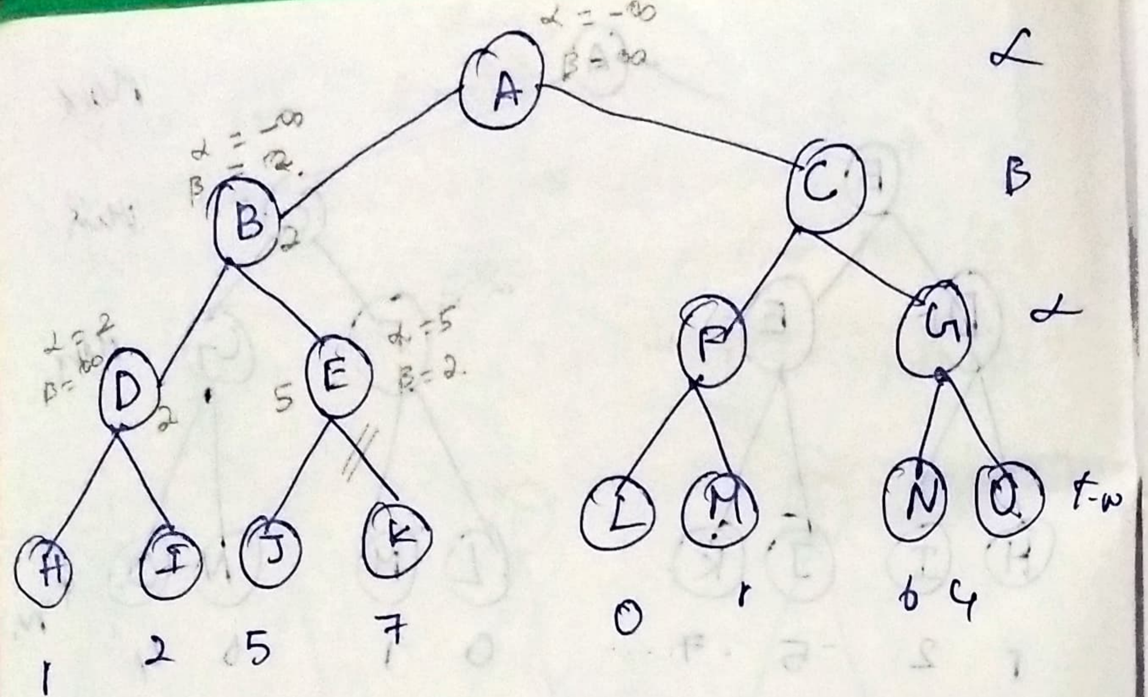
$(\infty, 2) = \beta = 2$



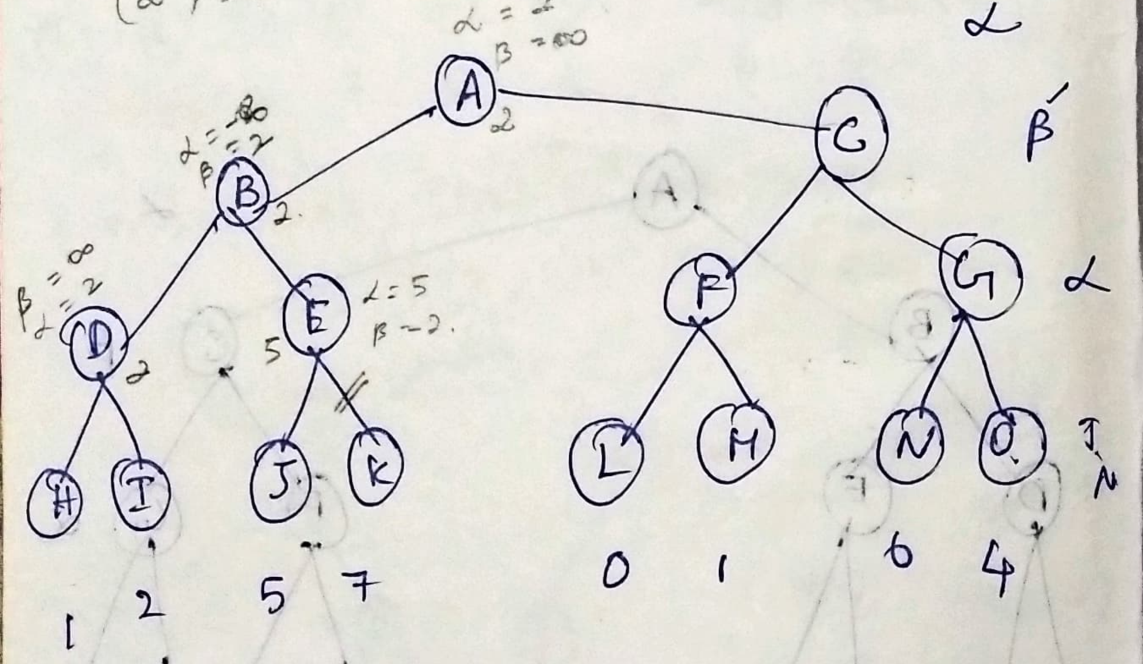
$(-\infty, 5) = \alpha = 5$

$\alpha > \beta$

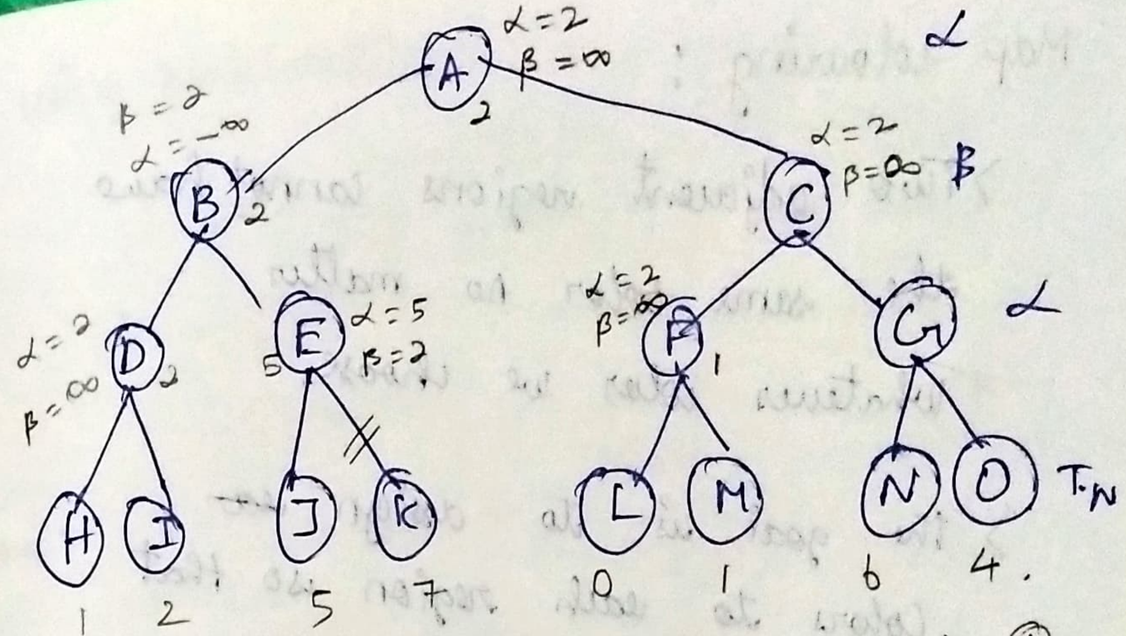
$5 > 2$



$(2, 5) = 2. = \beta.$

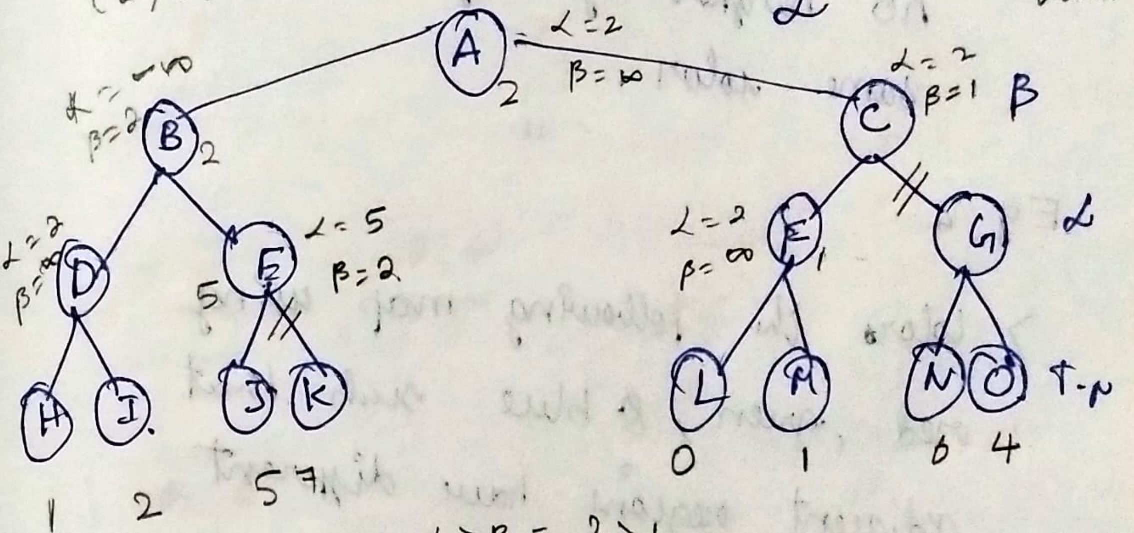


$(-\infty, 2) = \alpha = 2.$



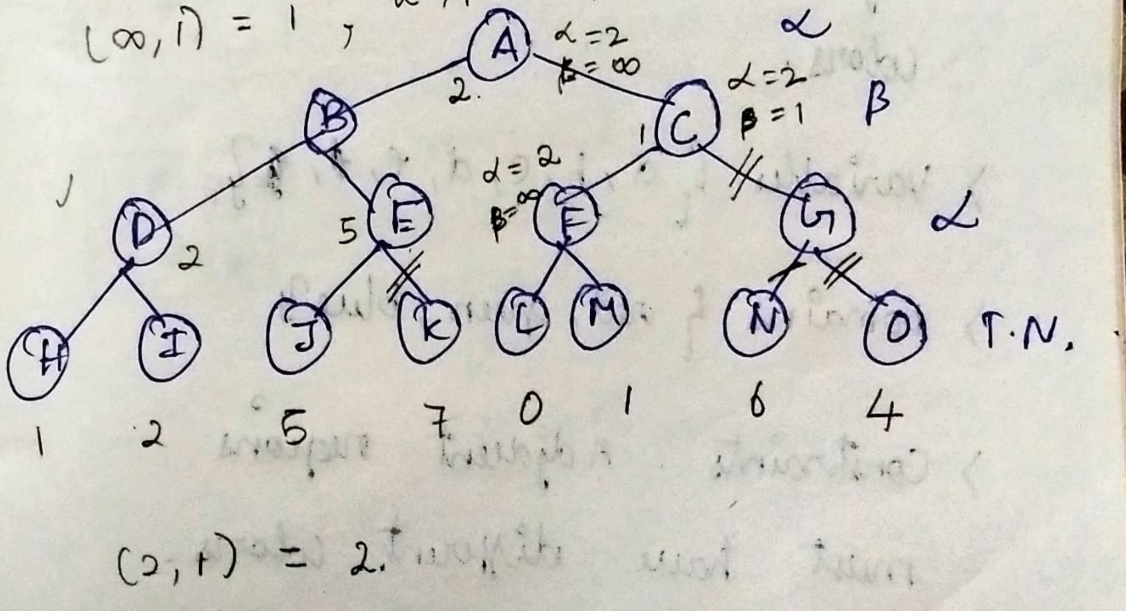
(2,0) = 2, (2,1) = 2, $\alpha=2$

(0,1) = 1 Node value.



$\alpha>\beta$ = 2>1

$(\infty,1)$ = 1,



(2,1) = 2.

$\alpha=2$.

Map colouring :

> Two adjacent regions cannot have the same color no matter whatever color we choose.

> The goal is to assign ~~sa~~ colors to each region so that the no neighboring regions have the same color.

Eg :

> Colors the following map using red, green, & blue such that adjacent regions have different colors.

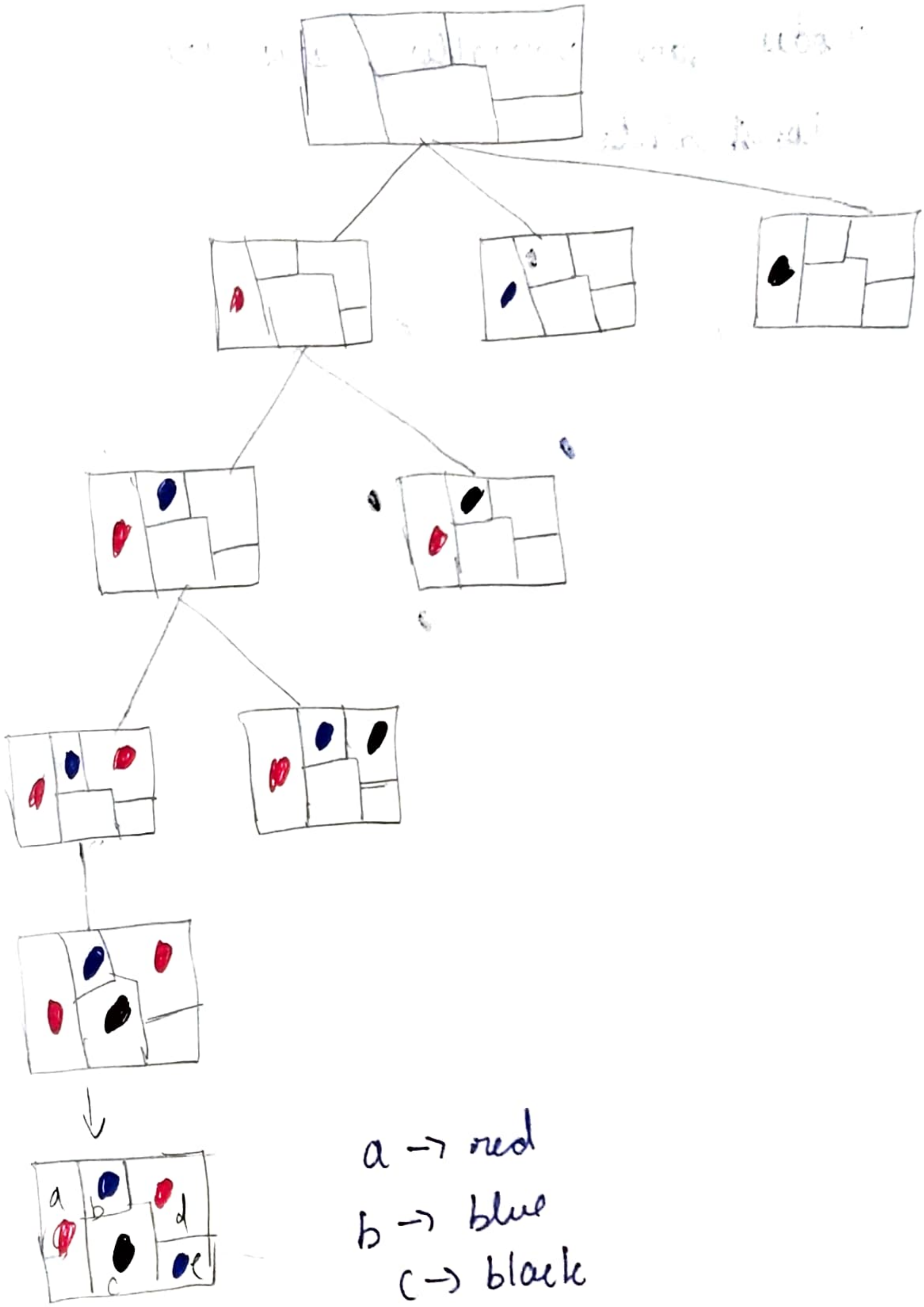
> Variables {a, b, c, d, e, ~~f~~, ~~g~~}.

> Domains {red, green, blue}

> Constraints : Adjacent regions must have different colors.

eg : a ≠ b

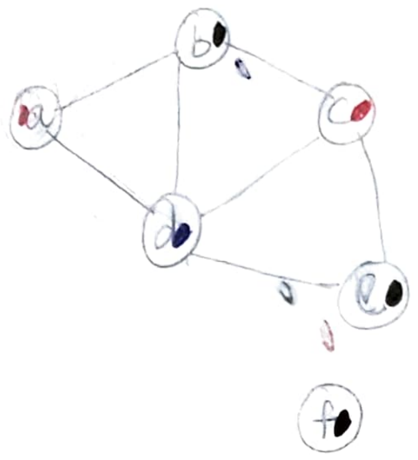
using backtracking :



a → red
b → blue
c → black
d → red.
e → Blue.

Constraint Graph :

nodes are variables, arcs are constraints.



Constraint Satisfaction Problem :

> constraint programming or constraint solving is about finding values for variables such that they satisfy a constraint (conditions).

> CSP = {V, D, C}.

Variables = V = {V1, ... Vn}

Domain = D = {D1, ... Dn}.

Constraints = C = {C1, ... CK}.

> Eg :

> Crossword puzzle

> Crypt-Arithmatic problem

> Map colouring problem.

Domain :

Discrete (Infinite, Finite)

Finite

Types of constraints in CSP :

Unary

binary

Linear

Non-Line

TO   Variables : T, O, G, U

+ GO   Domains : 0 - 9

OUT

	$\boxed{2}$	$\boxed{1}$
	$\boxed{8}$	$\boxed{1}$
$\boxed{1}$	$\boxed{0}$	$\boxed{2}$

T = 2

O = 1

G = 8

U = 0

S E N D

\+ M O R E

M O N E Y

	$\boxed{9}$	$\boxed{5}$	$\boxed{6}$	$\boxed{7}$
+	$\boxed{1}$	$\boxed{0}$	$\boxed{8}$	$\boxed{5}$
$\boxed{1}$	$\boxed{0}$	$\boxed{6}$	$\boxed{5}$	$\boxed{2}$

S = 9

M = 1

E = 5

O = 0

N = 6

R = 8

D = 7

Y = 2

$E + O = N$.

E's carry can be 0 or 1.

$0 \Rightarrow$. $\cancel{0} + E + \cancel{0} = N$.

$E \neq N$

$1 \Rightarrow$ $1 + E + \cancel{0} = N$.

$1 + E = N$. — ①.

$N + R = E$

carry can be 0 or 1.

$0 \Rightarrow$. $\cancel{0} + N + R = E$

$N + R = E + 10$

$1 + \cancel{E} + R = \cancel{E} + 10$.

$R \neq 9$

[∵ $E + 10$. why means the carry of the $E + O = 1$. so we add 10 to the E.]

$1 \Rightarrow$ $1 + N + R = E + 10$.

$1 + 1 + \cancel{E} + R = \cancel{E} + 10$

$R = 8$. ✓.

$5+6=11$ ✗

$5+7=12$ ✓

$6+7=13$ ✓

$D+E=Y+10$

$E=7 \Rightarrow 7.$

$E+O=N.$

$1+7=N.$

carry

$N \neq 8 \therefore \boxed{R=8}$

$D=7, E=6.$

$D+E=Y+10.$

$7+6=Y+10.$

$13=Y+10.$

$Y=3.$ ✗

$E+O=N.$

$1+6+0=N.$

$7 \neq N.$

$\boxed{D=7}$

$D=7, E=5.$

$E+O=N.$

$5+0=N.$

$1+5=6.$

$N=6.$ ✓

$$
\begin{array}{r}
E\ A\ T \\
+\ T\ H\ A\ T \\
\hline
A\ P\ P\ L\ E \\
\hline
\end{array}
$$

$$
\begin{array}{rrrrr}
 & & \boxed{} & \boxed{} & \boxed{} \\
 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\
\hline
\boxed{} & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\
\hline
\end{array}
$$