29/12

Practice Problem Set

1. 8 puzzle Problem

2	8	3
1	6	4
7		5

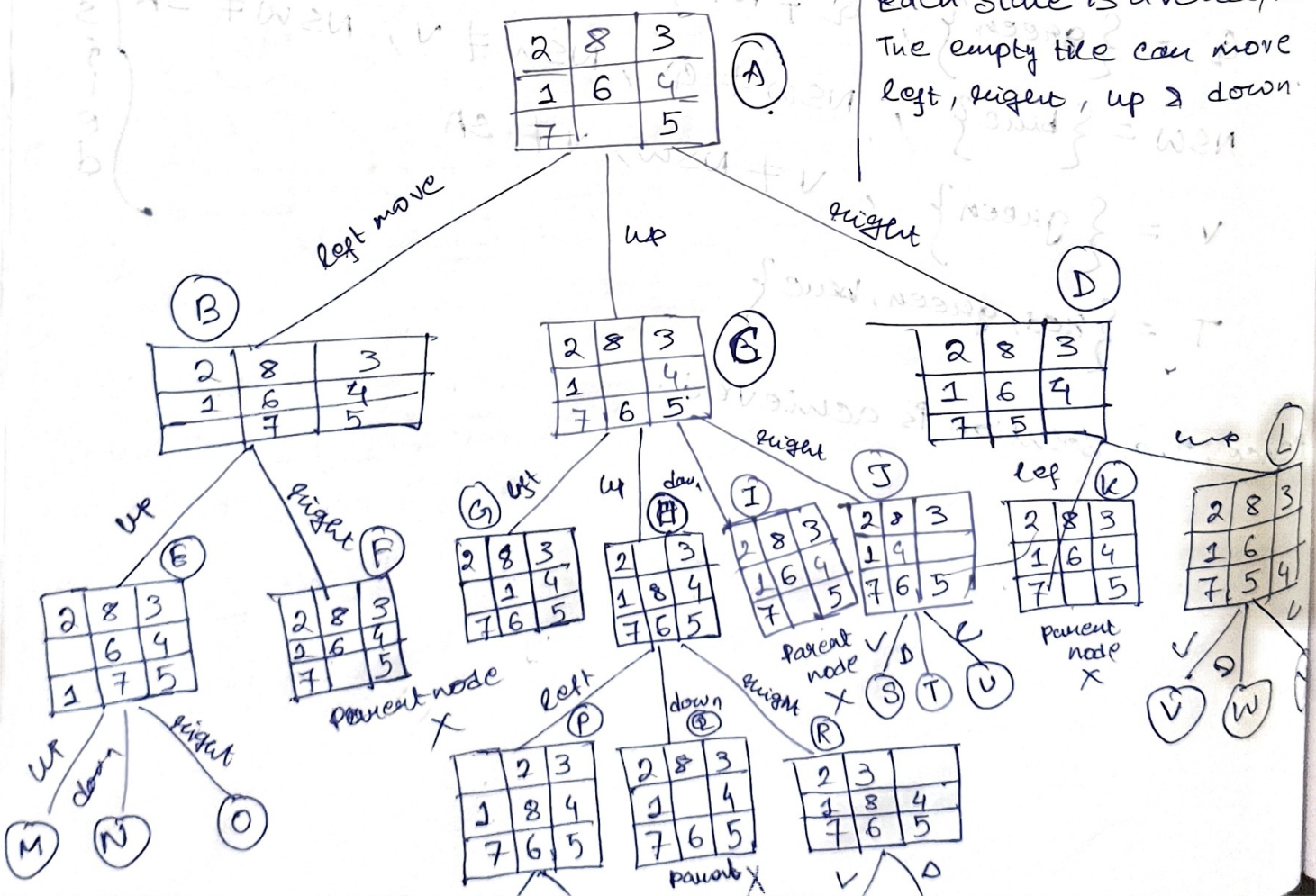
Initial State

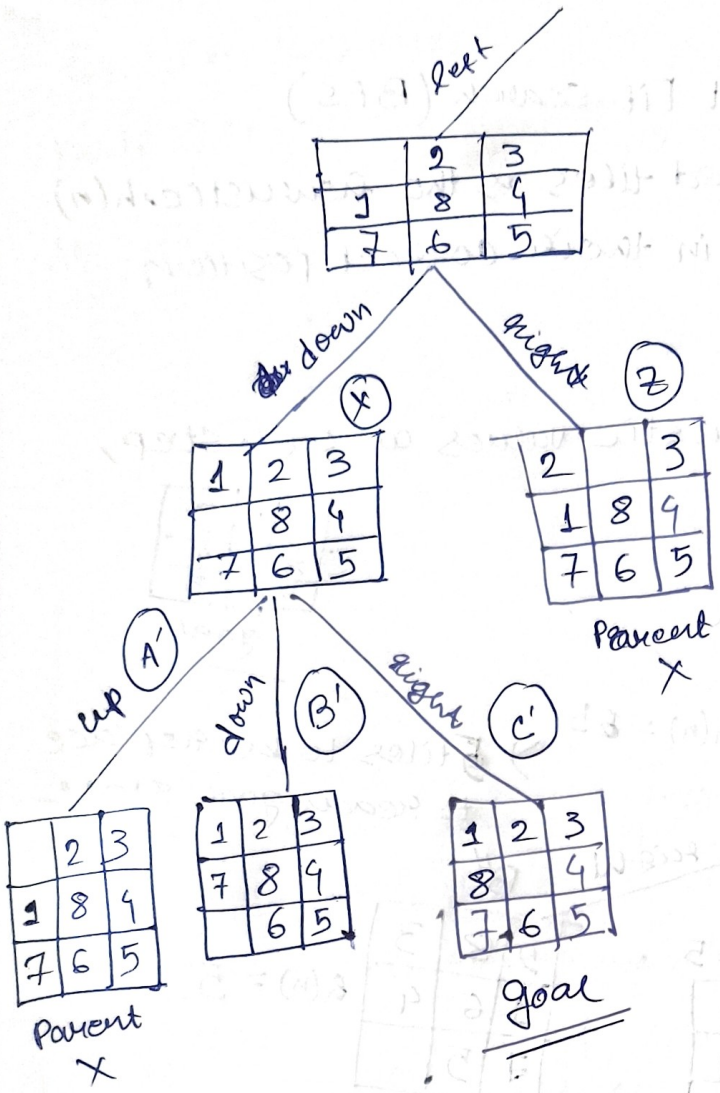
1	2	3
8		4
7	6	5

Goal State

Tasks.

1. Solve the problem using Uniform Cost Search. Use $g(n)$ as the cost of searching a node where each move has a uniform cost of 1.
   - Provide the solution path, the total cost, and the search tree.





The solution path:

$$A \xrightarrow[1]{} C \xrightarrow[1]{} H \xrightarrow[1]{} P \xrightarrow[1]{} X \xrightarrow[1]{} C'$$

The total cost, if every move cost $g(n) = 1$

T.C = 5 unit

Data Structure : Priority Queue.

Completeness : Complete

Optimality : Optimal

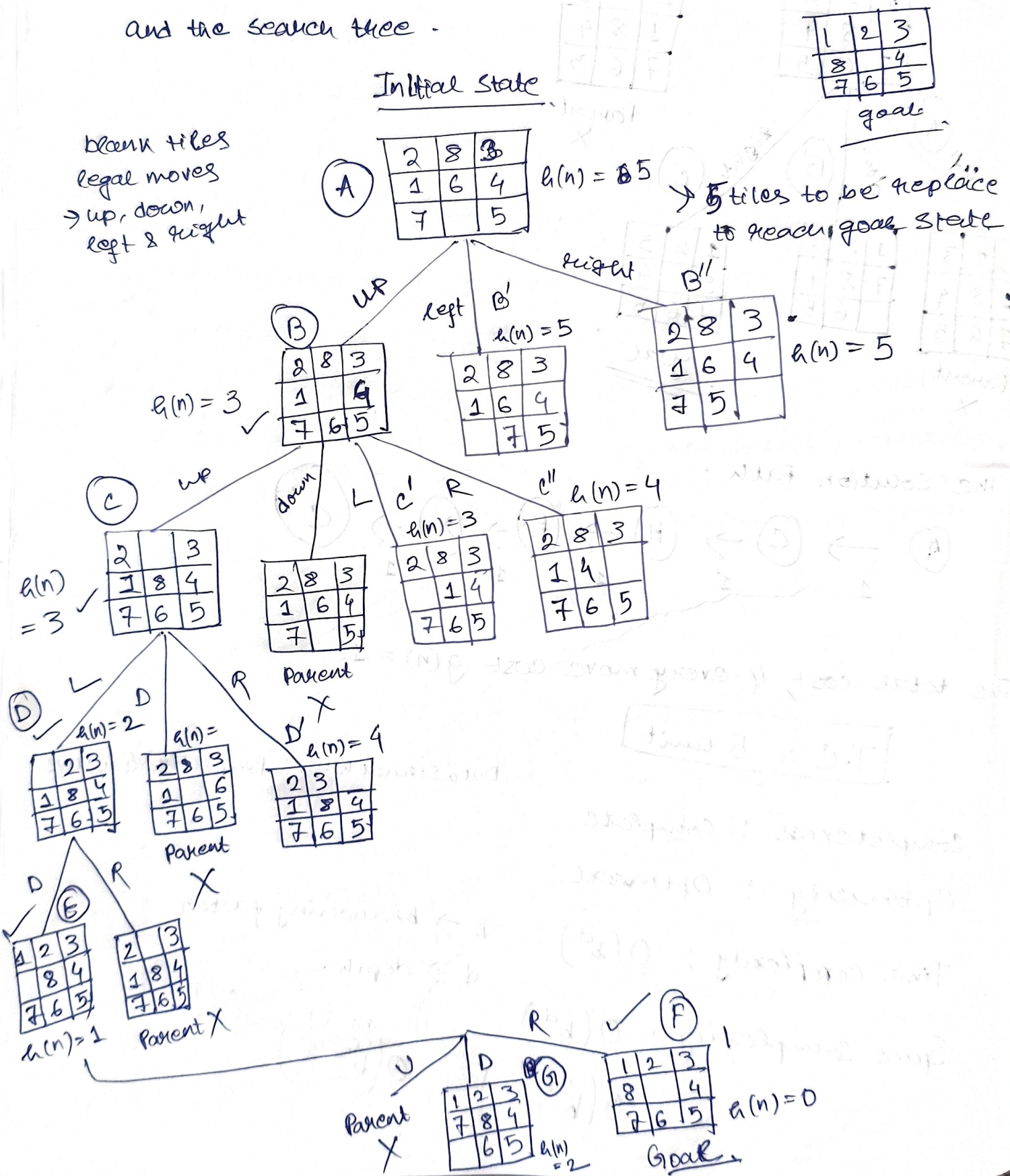
Time complexity : $O(b^d)$ $\quad b \rightarrow$ branching factor, $d \rightarrow$ depth.

Space complexity : $O(b^d)$ $\quad O\left(b^{C/\varepsilon + 1}\right)$ $\quad O\left(b^{C/\varepsilon + 1}\right)$ $\quad O(b^6)$

Tasks.

2. Solve the problem using Best Fit Search (BFS)

- Use the number of misplaced tiles as the heuristic $h(n)$ which counts the tiles not in their correct position (excluding the blank tile).
- Provide the solution path, heuristic values at each step, and the search tree.



Based on the misplaced tile heuristic value at each node $h(n)$ is calculated.

At initial state $h(n) = 5$

↓

2nd level

There are 3 nodes. $h(n) = 3$, $h(n) = 5$, $h(n) = 5$ → They will be inserted to priority queue based on $h(n)$.

3	5	5	

↓

3rd level

$h(n) = 3$, nodes will be popped out, goal test? NO

→ explore

There are 4 options, the empty tile can move up, down, left & right.

→ Down : parent node X can be ignored

→ up : $h(n) = 3$

→ left : $h(n) = 3$

→ Right : $h(n) = 4$

$h(n)$ values of the node to store.

PQ = | 3 | 3 | 4 | 5 | 5 |

up left left

popped out goal? NO → in visited list? NO → Explore

↓

4th level.

3 Legal moves possible

→ left : $h(n) = 2$

→ Down : Parent → ignore

→ Right : $h(n) = 4$

Visited list also to be maintained so that parent nodes can be ignored.

PQ {2, 3, 4, 4, 5, 5} ← corresponding nodes

popped out

PQ {1, 3, 4, 4, 5, 5}

↓

5th level.

2 legal moves

→ Down : $h(n) = 1$

→ Right : Parent ! ignore

visited-list {already visited nodes

PQ = {1, 3, 4, 4, 5, 5}

→ popped out.

goal? NO,

in visited list? NO

explore

↓

6th level.

3 Legal moved

→ up : parent : ignore

→ Down : $h(n) = 2$

→ right : $h(n) = 0$

} → Add to PQ

PQ = {0, 2, 3, 4, 4, 5, 5}

popped out

goal?

Yes

Goal Path:

A → B → C → D → E → F

$h(n) = 5$ $h(n) = 3$ $h(n) = 3$ $h(n) = 2$ $h(n) = 1$ $h(n) = 0$

Completeness : Complete

Optimality : Not guaranteed

Time complexity : $O(bd) \rightarrow O(b^d)$

Space " : $O(bd) \rightarrow O(b^d)$

b → branching factor

d → depth.

3. Solve the problem using A* search.

- Use the heuristic $h(n)$ as the no of misplaced tiles
- use $f(n) = g(n) + h(n)$, where $g(n)$ is the cost from the start node and $h(n)$ is the heuristic values.

Provide solution path, $g(n)$, $h(n)$ and $f(n)$ values for each step. and the search tree.

$g(A) = 0$, $h(A) = 5$

1st level

$f(A) = g(A) + h(A)$

$= 0 + 5 = 5$

A_5^0

2nd level

$f(B) = g(B^A) + h(B)$

$= g(A) + \text{cost } A \to B + h(B)$

$= 0 + 1 + 3 = 4$

B_4^A

$f(B') = g(B'^A) + h(B')$

$= g(A) + \text{cost } A \to B' + h(B')$

$= 0 + 1 + 5 = 6$

B_6^A

$f(B'') = g(B''^A) + h(B'')$

$= g(A) + \text{cost } A \to B'' + h(B'')$

$= 0 + 1 + 5 = 6$

B''^A_6

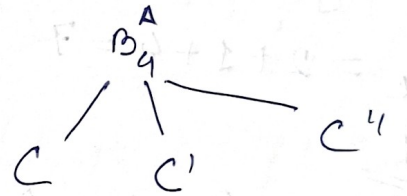
$PQ = \{B_4^A, B''^A_6, B'^A_6\}$

Visited_list $= \{A_5^0\}$

B_4

3rd Level

B_4^A → popped out → goal? NO → in visited list? NO → Explore



$f(C) = g(C^B) + h(C)$

$= g(B) + \text{cost } B \to C + h(C)$

$= 1 + 1 + 3 = 5$

C_5^B

$f(C') = g(C'^B) + h(C')$

$= g(B) + B \to C' + h(C')$

$= 1 + 1 + 3 = 5$

C'^B_5

$f(C'') = g(C''^B) + h(C'')$

$= g(B) + B \to C'' + h(C'')$

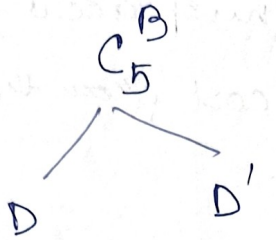
$= 1 + 1 + 4 = 6$

C''^B_6

PQ $\{C_5^B, C'^B_5, B'^A_6, B''^A_6, C''^B_6\}$

VL $\{A_5^0, B_4^A\}$

4th Level



$f(D) = g(D^C) + h(D)$

$= g(C) + C \to D + h(D)$

$= 2 + 1 + 2 = 5$

D_5^C

$f(D') = g(D'^C) + h(D')$

$= g(C) + C \to D' + h(D')$

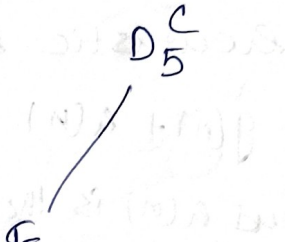
$= 2 + 1 + 4 = 7$

D'^C_7

PQ $\{ D_5^C, C_5'^B, B_6'^A, B_6''^A, C_6''^B \}$

VL $\{ A_5^0, B_4^A, C_5^B \}$

5th Level



$f(E) = g(E^D) + h(E)$

$= g(D) + \text{cost } D \to E + h(E)$

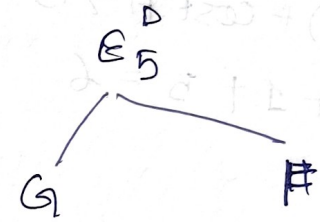
$= 3 + 1 + 1 = 5$

E_5^D

PQ $\{ E_5^D, C_5'^B, B_6'^A, B_6''^A, C_6''^B \}$

VL $\{ A_5^0, B_4^A, C_5^B, D_5^C \}$

6th Level.



$f(G) = g(G^E) + h(G)$

$= g(E) + E \to G + 2$

$= 4 + 1 + 2 = 7$

G_7^E

$f(F) = g(F^E) + h(F)$

$= g(E) + E \to F + 0$

$= 4 + 1 + 0 = 5$

F_5^E

PQ $\{ F_5^E, C_5^B, B_6'^A, B_6''^A, C_6''^B \}$

VL $\{ A_5^0, B_4^A, C_5^B, D_5^C, E_5^D \}$

→ F_5^E popped out

goal? Yes

path to Goal :

$$A_5^0 \rightarrow B_4^A \rightarrow C_5^B \rightarrow D_5^A \rightarrow E_5^D \rightarrow F \text{ (Goal Node)}$$

Completeness : Complete

Optimality : Optimal for admissible heuristic

Time Comp : $O(b^d)$

Space " : $O(b^d)$

Impact of heuristic on BFS & A*

Use of heuristic function $h(n)$ in in BFS and A* has made to get the solution quicker.

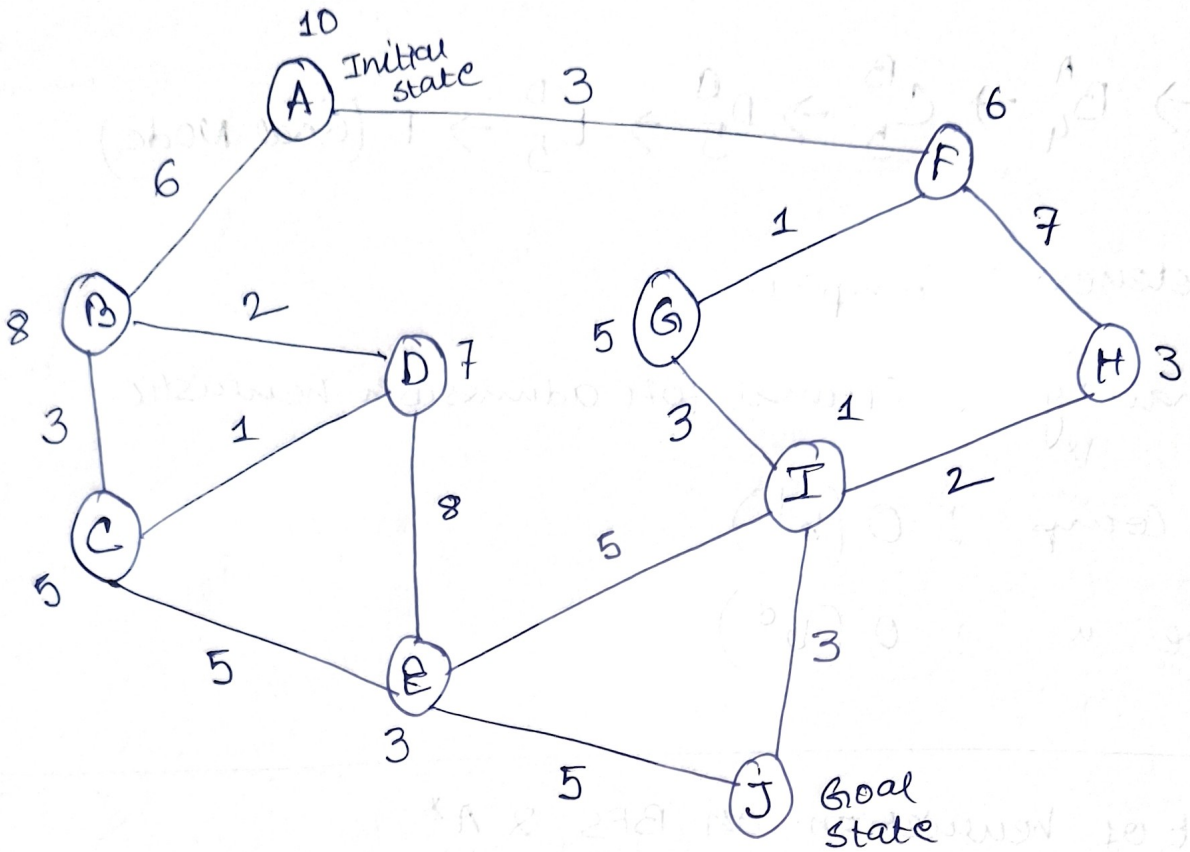
in UCS, search is done by conventional Breadth First Search systematically. As every path cost unit is same, time comp. will take $O(b^d)$...

But because of using h values those nodes are getting priority whose heuristic values are less.

Hence, if heuristic $\leq$ original cost i.e admissible heuristic it will be an optimal solution in quickest time.

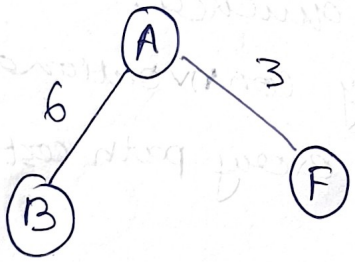
Best Time comp $O(V)$ vertices.

Worst " " $O(b^d)$.



Uniform Cost Search (UCS)

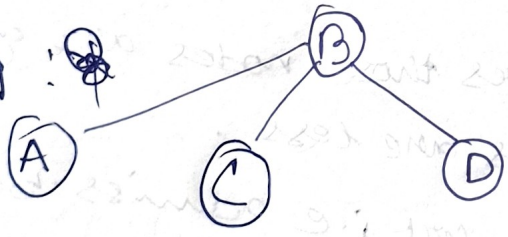
Level 1 :



$PQ = \{ B_6^A, F_3^A \} \{ F_3^A, B_6^A \}$

$VL \{ A_0^\circ \}$

Level 2 :



B_6^A popped → goal ? NO → in VL ? → Explore ← NO

$g(A^B) = g(B) + \text{cost } B \to A = 6 + 6 = 12$ → A_{12}^B → Already in VL with better $g(n)$

$g(C^B) = g(B) + \text{cost } B \to C = 6 + 3 = 9$ → C_9^B

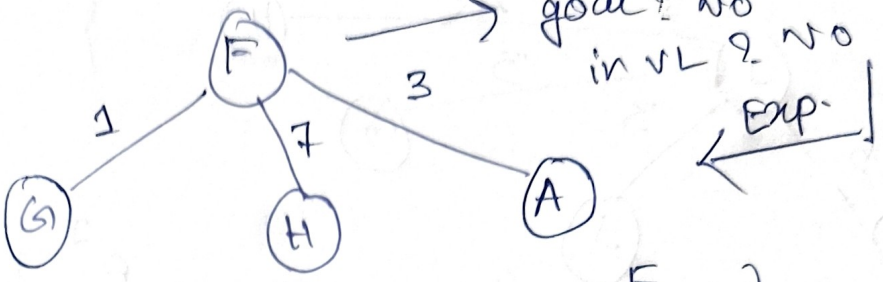
$g(D^B) = g(B) + \text{cost } B \to D = 6 + 2 = 8$ → D_8^B

→ Add to PQ

$PQ \{ I_7^G, D_8^B, C_9^B, H_{10}^F \}$

$VL \{ A_0^\circ, F_3^A, G_4^F, B_6^A \}$

Level 2



$g(G)^F = g(F) + F \to G = 3 + 1 = 4$, G_4^F

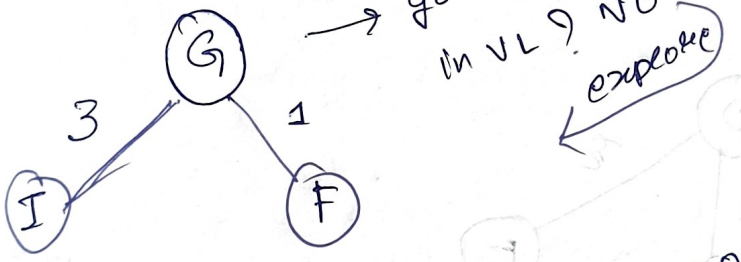
$g(H)^F = g(F) + F \to H = 3 + 7 = 10$, H_{10}^F } → Add to PQ

$g(A)^F = g(F) + F \to A = 3 + 3 = 6$, A_6^F → already in VL with a better g(n), ignore

PQ $\{ G_4^F , B_6^A , H_{10}^F \}$

VL $\{ A_0^\circ , F_3^A \}$

Level 3 :



$g(I^G) = g(G) + \text{cost } G \to F$

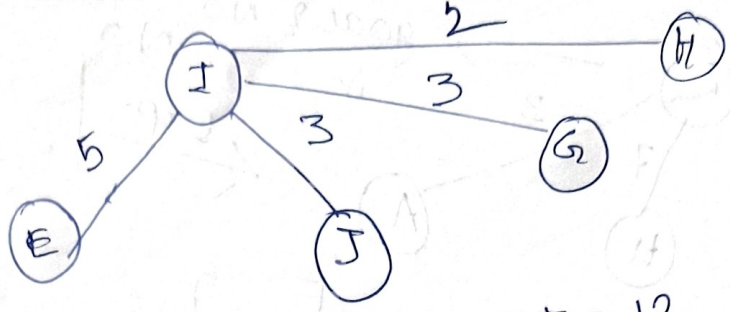
$= 4 + 3 = 7$, I_7^G → add to PQ

$g(F^G) = 3 + 3 = 6$, F_6^G → already in VL with a better g(n)

PQ $\{ B_6^A , I_7^G , H_{10}^F \}$

VL $\{ A_0^\circ , F_3^A , G_4^F \}$

Level 5



$g(E^I) = g(I) + I \to E = 7 + 5 = 12$, E^I_{12}

$g(J^I) = g(I) + I \to J = 7 + 3 = 10$, J^I_{10} } Add to PQ.

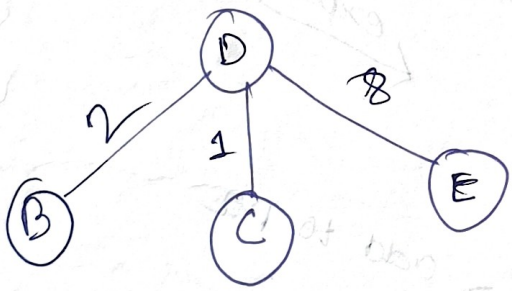
$g(G^I) = g(I) + I \to G = 7 + 3 = 10$, G^I_{10} → in VL with a better g(n)

$g(H^I) = g(I) + I \to H = 7 + 2 = 9$, H^I_9 → in PQ, but this a better g(n) replace.

PQ $\{ D^B_8 , C^D_9 , H^I_9 , J^I_{10} , E^I_{12} \}$

VL $\{ A^\circ_0 , F^A_3 , G^F_4 , B^A_6 , I^G_7 , H^F_{10} \}$

Level 6.



$g(B^D) = g(D) + D \to B = 8 + 2 = 10$, B^D_{10} → in VL with a better g(n)

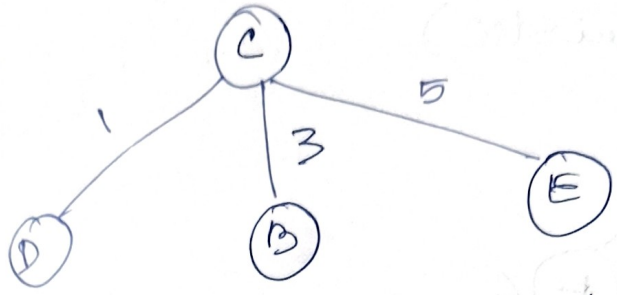
$g(C^D) = g(D) + D \to C = 8 + 1 = 9$, C^D_9 → in PQ with same g(n)

$g(E^D) = g(D) + D \to E = 8 + 8 = 16$, E^D_{16} → in PQ with a better g(n)

PQ $\{ C^B_9 , H^I_9 , J^I_{10} , E^I_{12} \}$

VL $\{ A^\circ_0 , F^A_3 , G^F_4 , B^A_6 , I^G_7 , H^F_{10} , D^B_8 \}$

Level 8



$g(D^C) = g(C) + C \to D = 9 + 1 = 10$ / D^C_{10} → in VL

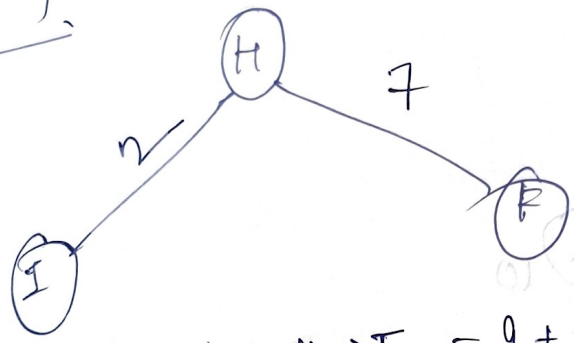
$g(B^C) = g(C) + C \to B = 9 + 3 = 12$ / B^C_{12} → in VL

$g(E^C) = g(C) + C \to E = 9 + 5 = 14$ / E^C_{14} → in PQ

} ignore

$PQ \{ H^I_9, J^I_{10}, E^I_{12} \}$ $VL = VL + C^B_9$

Round 9.



$g(I^H) = g(H) + H \to I = 9 + 2 = 11$, I^H_{11} in VL

$g(F^H) = g(H) + H \to F = 9 + 7 = 16$, F^H_{16} ← in VL

} ignore

$PQ \{ J^I_{10}, E^I_{12} \}$, $VL = VL + H^I_9$

Round 10

J → goal? Yes

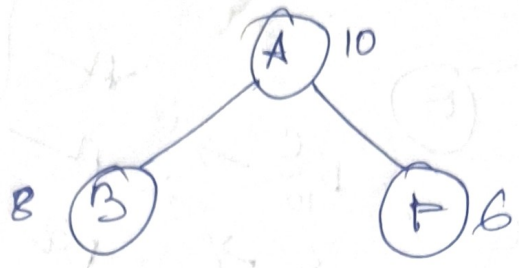
Path to Goal.

$A \xrightarrow{3} F \xrightarrow{1} G \xrightarrow{3} I \xrightarrow{3} J$

Total path cost = $3 + 1 + 3 + 3 = 10$

Best Fit Search (Heuristic)

Round 1



$h(A) = 10$
$h(B) = 8$
$h(F) = 6$

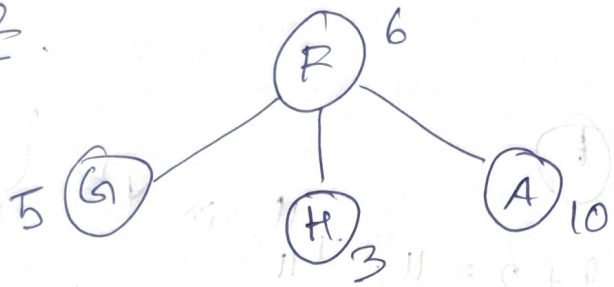
$PQ\{A_{10}^{0}\}$, $VL = \{\}$

↙ pop

$PQ\{F_6^A, B_8^A\}$, $VL\{A_{10}^{0}\}$

↙ pop

Round 2



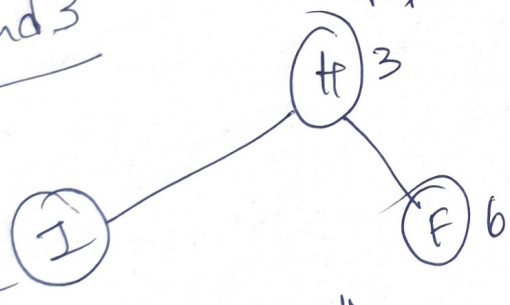
$h(G) = 5$
$h(H) = 3$

$PQ\{H_3^F, G_5^F, B_8^A\}$

$VL\{A_{10}^{0}, F_6^A\}$

↘ pop

Round 3



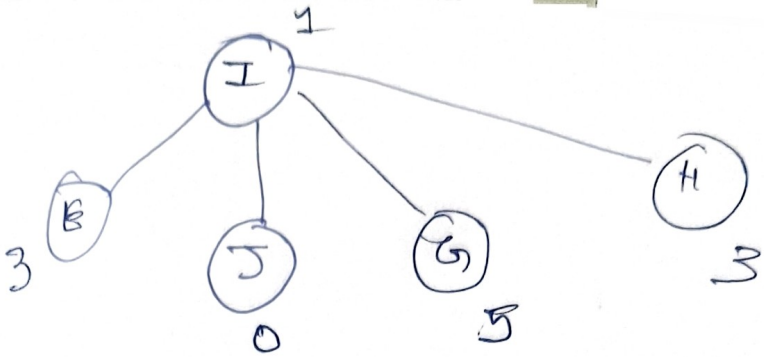
$h(I) = 1$, I_1^H → PQ add

$h(F) = 6$, F_6^H → in VL, ignore

$PQ\{I_1^H, G_5^F, B_8^A\}$

$VL\{A_{10}^{0}, F_6^A, H_3^F\}$

Round 4



$h(J) = 0$, J_0^I } PQ addition
$h(E) = 3$, E_3^I }
$h(G) = 5$, G_5^I → in PQ
$h(H) = 3$, H_3^I → in VL

PQ { J_0^I , E_3^I , G_5^F , B_8^A }

VL { A_{10}^0 , F_6^A , H_3^F , I_1^H }

Round 5

(J) → goal node? Yes.

PATH to goal.

$A_{10}^0 \rightarrow F_6^A \rightarrow H_3^F \rightarrow I_1^H \rightarrow J_0^I$

(Start Node) A → F → H → I → J (Goal state)
10 6 3 1 0

Note: Much faster than UCS.