Q 1

BFS	DFS
1) BFS finds the shortest path to the destination	1) DFS goes to the bottom of a subtree, then backtracks
2) The full form of BFS is Breadth first search	2) The full form of DFS is Depth first search
3) It uses a queue to keep track of the next location to visit	3) It uses a stack to keep track of the next location to visit.
4) BFS traverse according to tree level	4) DFS traverses according to tree depth.
5) It is implemented using FIFO list.	5) It is implemented using LIFO list.
6) It requires more memory as compare to DFS	6) It require less memory as compare to BFS.
7) This algorithm gives the shallowest path solution.	7) This algorithm doesn't guarantee the shallowest path sol^
8) There is no need to back-tracking in BFS	8) There is need of backtracking in DFS.
9) You can never be trapped into infinite loop	9) You can be trapped into infinite loops.

2023/10/15 14:32

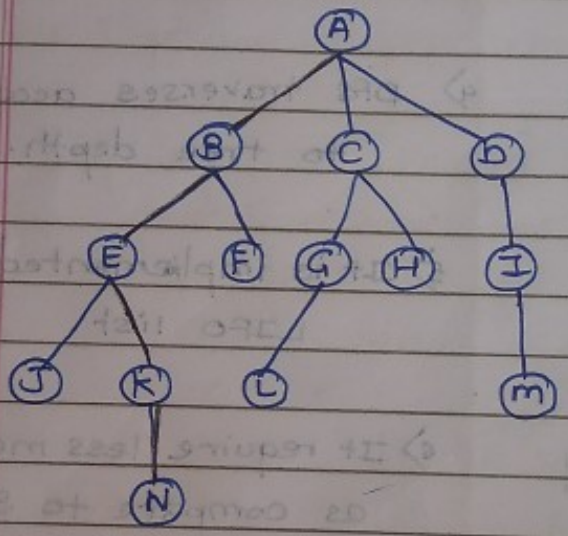
10) If you do not find any goal, you may need to expand many nodes before the soln is found.

10) If you do not find any goal the leaf node backtracking may occur.

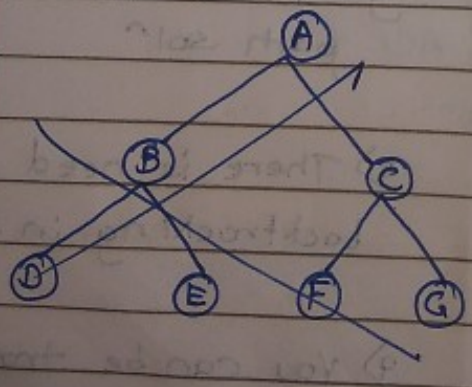
Q 2 Uninformed Search –

Q 2 a) Breadth First Search –

- Uninformed Search technique.
- It uses FIFO method
- It moves level by level



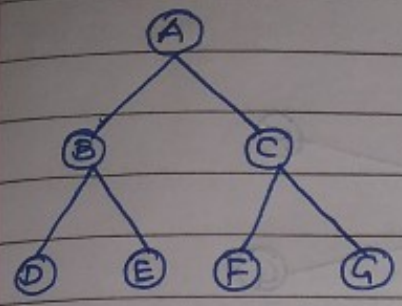
~~A~~
~~B~~CD
~~C~~DEF
~~D~~EFGH
~~E~~FGHI
~~F~~GHIJK
~~G~~HIJK
~~H~~IJKL
~~I~~JKL
~~J~~KLM
~~K~~LM
KM~~N~~
~~M~~N
N



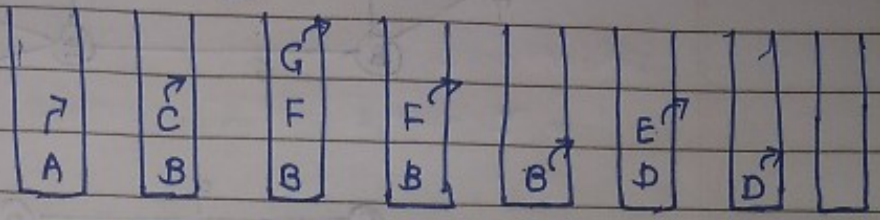
		C	G
A		B	F
			B

2) Depth first search

Ex.



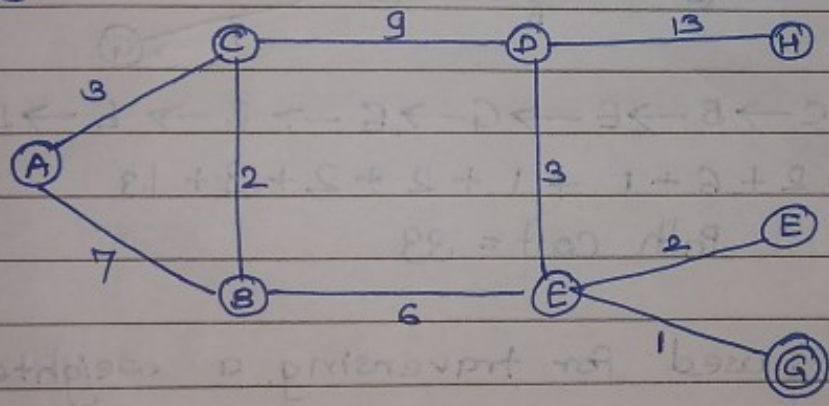
A → | C | B | or | B | C |

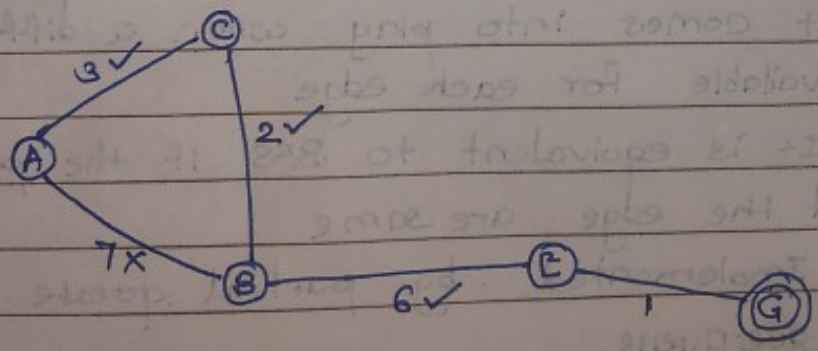


Final path = ACGFBED

3) Uniform Cost search

Ex 1-





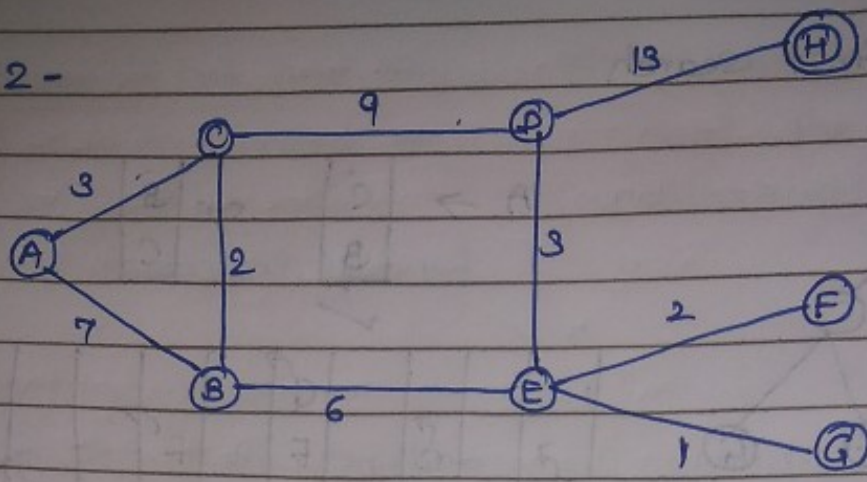
A → C → B → E → G

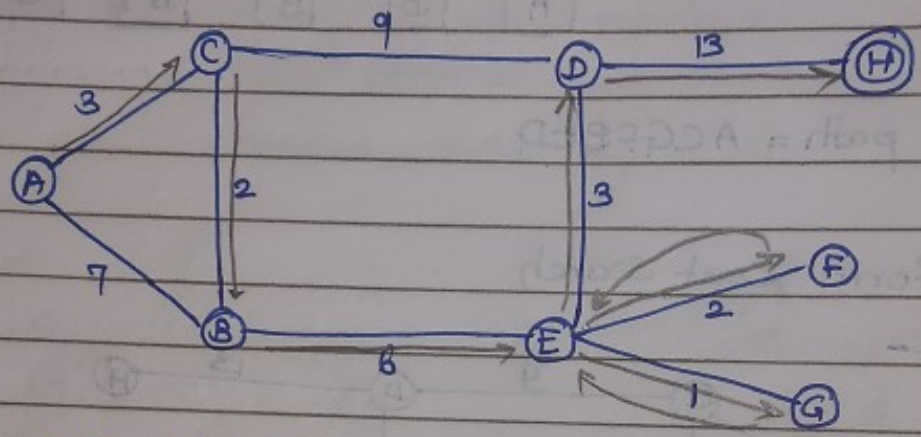
3 + 2 + 6 + 1 = 12

Path cost = 12

2023/10/15 14:32

Ex 2 -





A → C → B → E → G → E → F → E → D → H

3 + 2 + 6 + 1 + 1 + 2 + 2 + 3 + 13

Path cost = 33

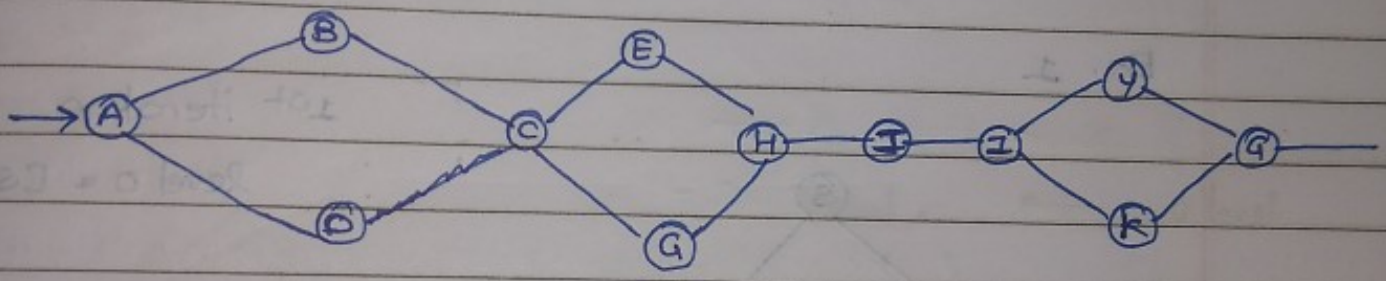
- It is used for traversing a weighted tree or graph
- It comes into play when a diff cost is available for each edge
- It is equivalent to BFS if the path cost of all the edge are same
- Implemented by ~~partial queue~~. priority que: queue
- used to solve any graph or tree
- Highest priority to minimum cost backtracking is there

2023/10/15 14:32

Informed Search -

4) Bidirectional Search

> Two diff. searches run simultaniously
  1) Forward Search
  2) Backward Search
> Single search graph replace with Two small graph.
> Any Search Technique can be used (BFS, DFS)
> Search Stop condition (when graph intersect with each other)



Forward	Backward
A	G
ABD	GKY
ABDC	GKYI
ABDCEF	GKYII
/	
ABDCEFH	GKYIIH

A → B → D → C → E → F → H → I → I → Y → K → G

2023/10/15 14:32

5) Iterative deepning search

> It is combination of both BFS & DFS
> Best depth limit is found out by gradually increasing limit

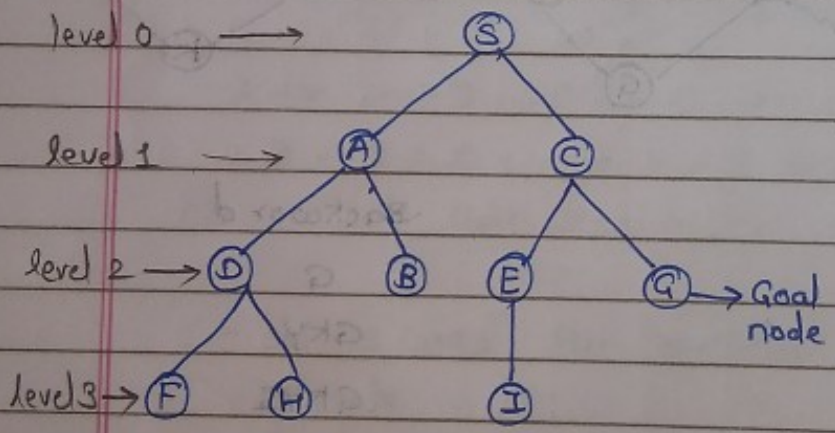
Advantages

> Fast & Less memory requires.

Disadvantages

> Repeat the work / process

Ex. 1



1st iteration

level 0 = [S]

2nd iteration

level 0 + level 1 = [S → A → C]

3rd iteration

level 0 + level 1 + level 2 =

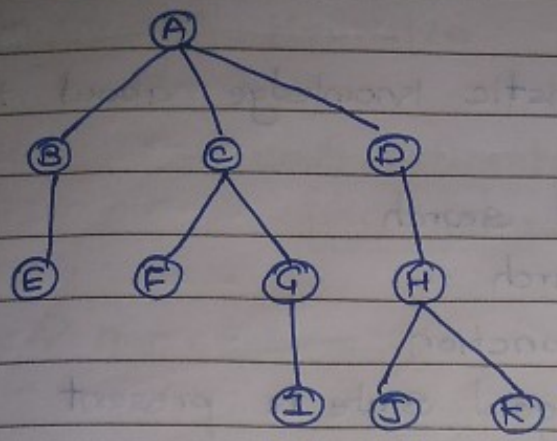
[S → A → D → B → C → E → G]

4th iteration

~~level 0 + level 1 + level 2 + level 3 =~~

~~[S → A~~

Ex 2.

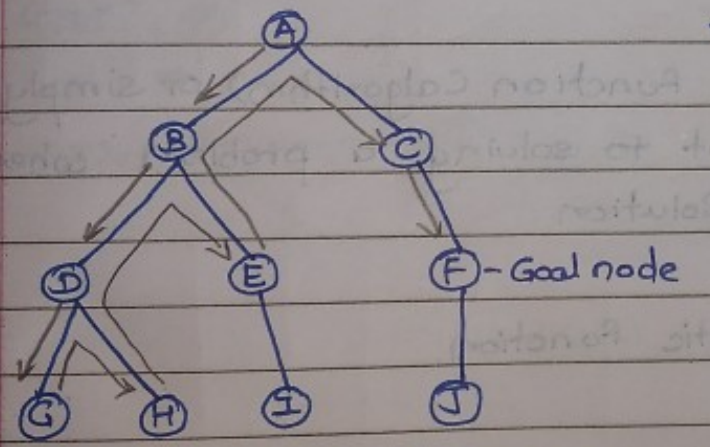


Depth level	Iterative deepning
0	A
1	ABCD
2	ABECFGDH
3	ABECFGIDHJK

Goal Node = K

6) Depth limited search -

Ex 1



F - Goal node

Path = A → B → D → G → H → E → C → F

Depth limit = 2

- Depth limited search algorithm is similar to depth-first search with a predetermined limit
- Depth limited search is memory efficient.
- Depth limited search also has disadvantage of incompletness.
- It is not optimal if the problem has more than one sol?

2023/10/15 14:32

Informed Search

> Here we use heuristic knowledge about this problem.

> Types of Informed search
  1. Heuristic search
  2. Heuristic function

> Information about goal state is present

> Find optimal or good sol^n

Heuristic search form optimize path
Heuristic function a function which will give us a sol^n

~~Best first search~~

Heuristic function -

A Heuristic function (algorithm) or simply a heuristic is a shortcut to solving a problem when there is no exact solution

Example of heuristic function

1) 8 queen problem

7	2	4
5		6
8	3	1

	1	2
3	4	5
6	7	8

h_1 = Total moves = 8

$h_2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$

Total moves = 18 + 8 = 26

2023/10/15 14:32

Types of Heuristic

1) Admissible

- underestimate cost

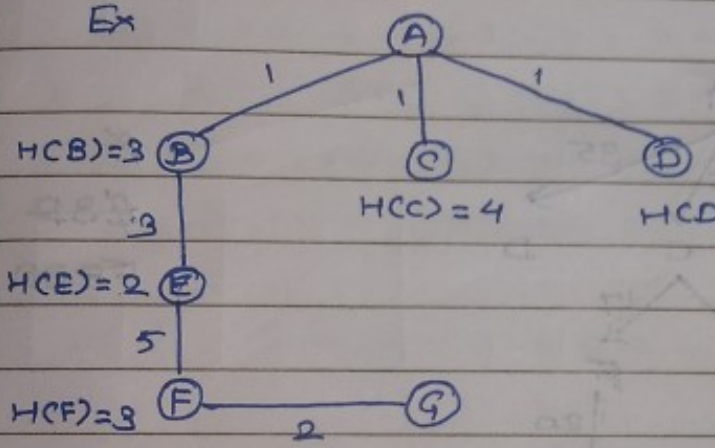
$$H(n) \leq H'(n)$$

2) non - Admissible

- Overestimated Cost

$$H(n) > H'(n)$$

Ex



Heuristic value of B

H(B) = 3

Cost value of A to B = 1

$f(n) = H(n) + G(n)$

B = 4 , C = 5 D = 6

Actual Cost = 11

H(B) = 3

3 < 11

1) Best first Search

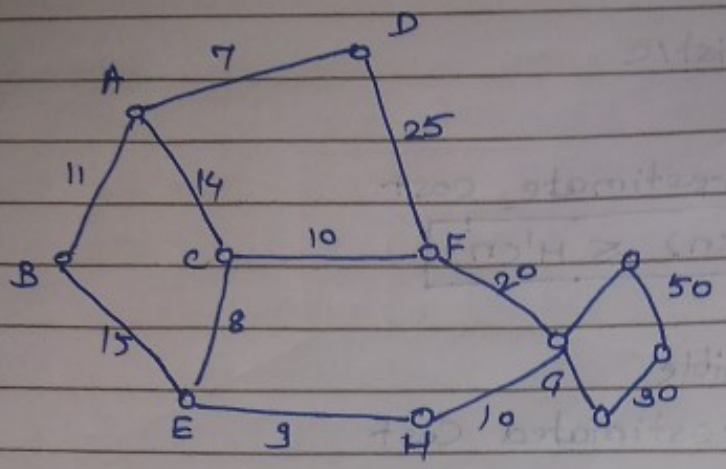
- It always selects the path which appers best at that movement
- Combination of Both DFS & BFS
- It uses heuristic function $h(n) <= h^*(n)$

$h(n)$ = heuristic cost

$h^*(n)$ = estimated cost

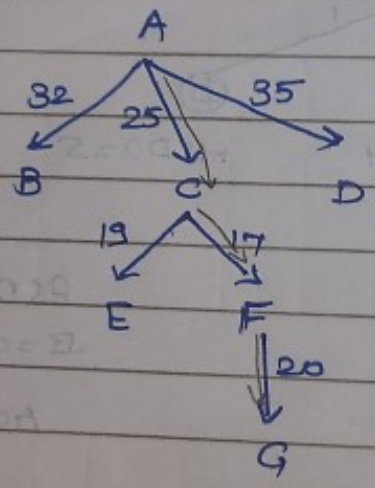
- The greedy best first algorithm implemented by priority queue.

2023/10/15 14:32



straight line dist.

A → G = 40
B → G = 32
C → G = 25
D → G = 35
E → G = 19
F → G = 17
H → G = 10
G → G = 0



~~A~~
~~CBD~~
~~FEBD~~

Final path = ACFG

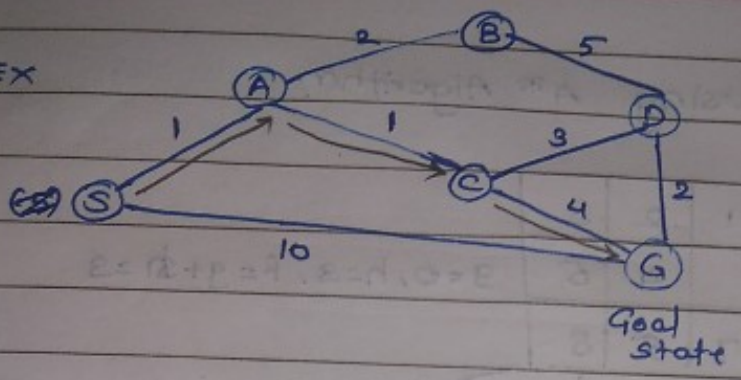
2) A* search

> f(n) = g(n) + h(n)

g(n) = actual cost from start node to n
h(n) = estimation cost from n to goal node

> A* search is both optimal & complete

Ex



state	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

① $S \to A \Rightarrow f(n) = g(n) + h(n) = 1 + 3 = 4$ ✓

$S \to G = f(n) = 10 + 0 = 10$

② $S \to A \to B \Rightarrow f(n) = 3 + 4 = 7$

$S \to A \to C \Rightarrow f(n) = 2 + 2 = 4$ ✓

③ $S \to A \to C \to D \Rightarrow f(n) = 5 + 6 = 11$

$S \to A \to C \to G \Rightarrow f(n) = 6 + 0 = 6$ ✓

$S \to A \to C \to G = 1 + 1 + 4 = 6$

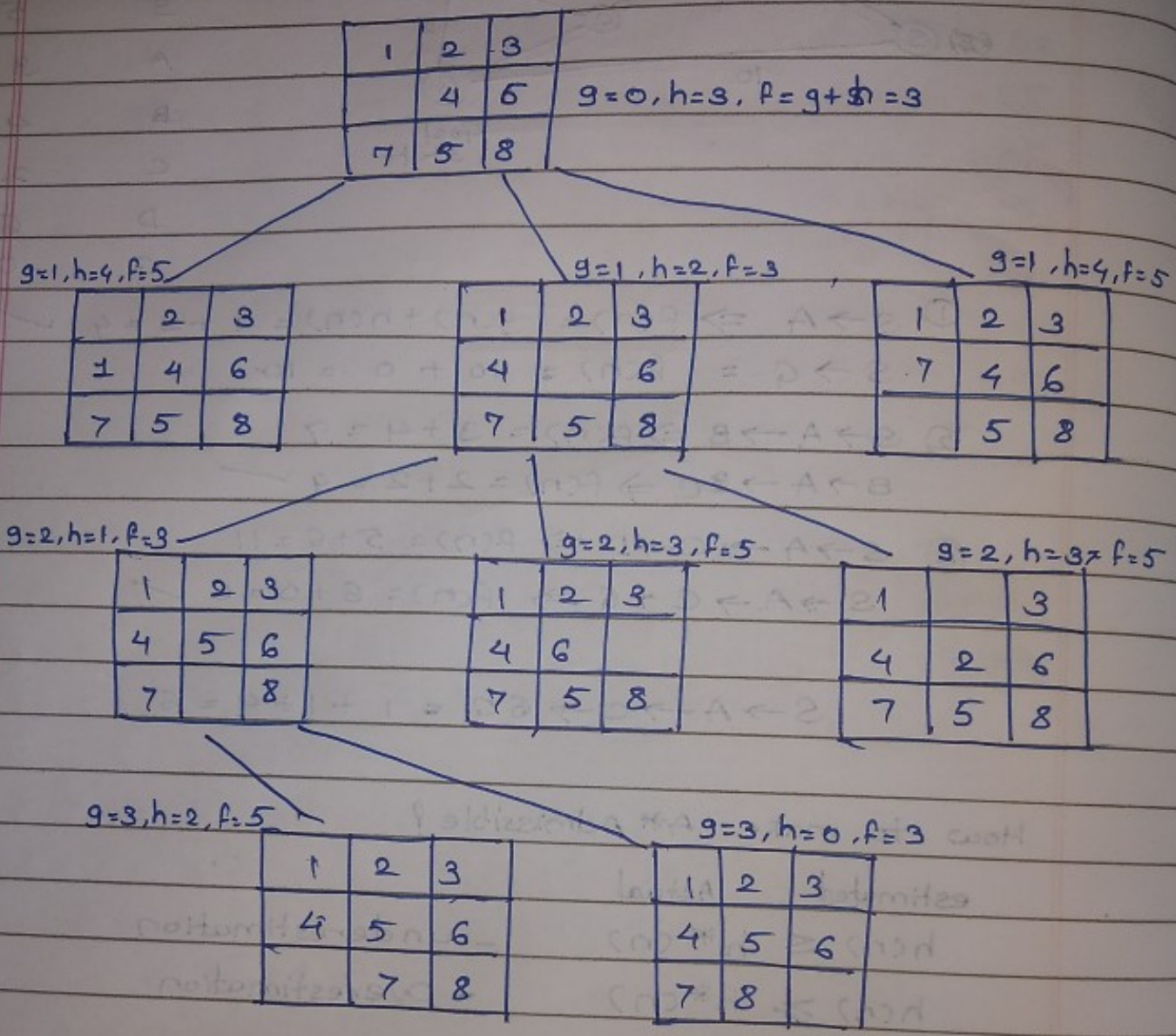
How to make A* Admissible?

estimated Actual

$h(n) \leq h^*(n)$ - underestimation

$h(n) \geq h^*(n)$ - Overestimation

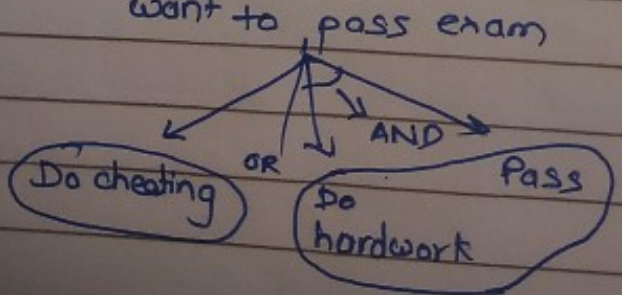
* Solving 8-puzzle using A* Algorithm



③ AO* Algorithm -

AND / OR - special graph →+

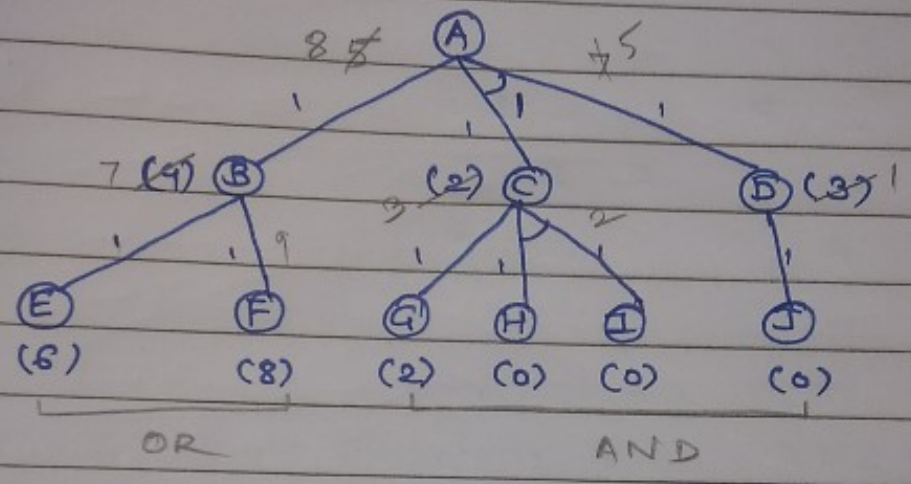
- Problem Decomposition (Breakdown into small pieces)

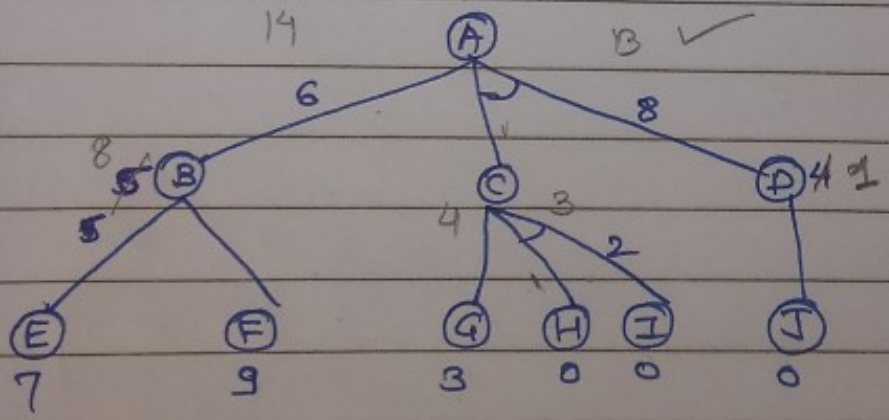


2023/10/15 14:33

- It the basis of heuristic value we find the optimal solution
- informed search technical.
- guarranted optimal sol^n

Ex





2023/10/15 14:33

* Solve the following crypt arithmetic problem

```
  C R O S S
+ R O A D S
-----------
D A N G E R
```

→

```
  C5 C4 C3 C2 C1
    C R O S S
+   R O A D S
-------------
D A N G E R
```

$C_5 = 1$ becoz C_5 is carry

$S + S = R$ so R must be even no so possibility

$\{0, 2, 4, 6, 8\}$

Eg. ● If we take 0 is value so

$S + S = R$

$0 + 0 = 0$

S & R value must be same so it create conflict

If we take 2 is value then

$S + S = R$

$(2 + 2 = 4)$

But then the conflict occure with A & R

If we take ~~3 is value~~ $R = 6$, $S = 3$

$S + S = R$

$3 + 3 = 6$

	C 9	R 6	O 2	S 3	S 3
+	R 6	O 2	A 5	D 1	S 3
D 1	A 5	N 8	G 7	E 4	R 6

* Solve the following crypt Arithmetic problems

```
   B  A  S  E
+  B  A  L  L
-------------
G  A  m  E  S
```

→

E + L = S → (a) [no carry]

S + L = E + 10 → (b) [Carry]

↳ E = S - L + 10

S + L = E ↓

S + L = S - L + 10

or 2L = 10

L = 5

So E + L = S

5 = S - E

assing value

so we find the combination of S & E ① (5,0) × L

(6,1) × 9

(7,2)

(8,3)

(9,4)

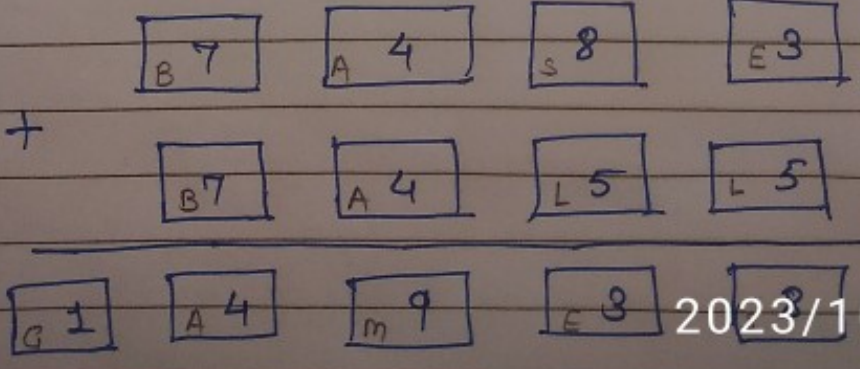
B + B = A + Carry

value of B must be 5, 6, 7, 8, 9

Let (S, E) = (7, 2) & B = 6 B + B = 12 (A) → c value already assign

(c)

Let (S, E) = (8, 3) & B = 7 B + B = 14



2023/10/15 14:33