DATE

Artificial intelligence

Tasks

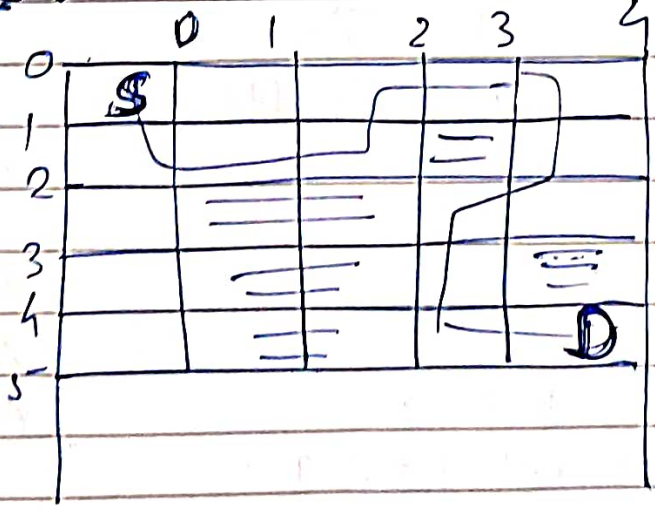
- formal task → Games mathematics
- Expert task → scientific analysis.

State space search

Elements :- (i) states :- initial
final

(ii) set of Rules

Example :-

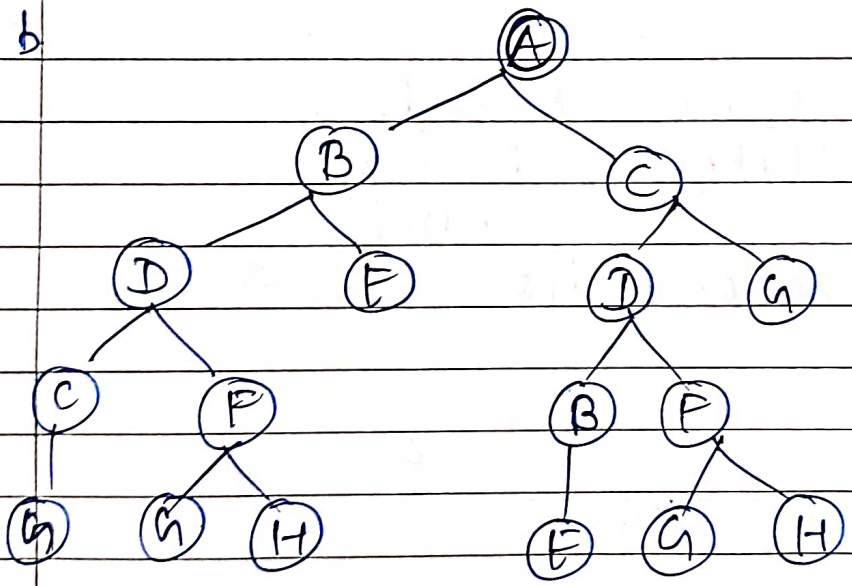


DATE

Breadth first search

1. Create a variable called Node list and set it to initial state

2. Until a goal is not reached :-
   a. Remove the first element from Node list and call it E.
      If Node list is empty, then quit
   b.



Node list: A — A
Node list: B C — A B
Node list: D E C — A B C
Node list: ~~F E E~~ D.

DATE

Advantage of BFS

- simplest search algorithm
- BFS gives guaranteed results

Disadvantage of BFS

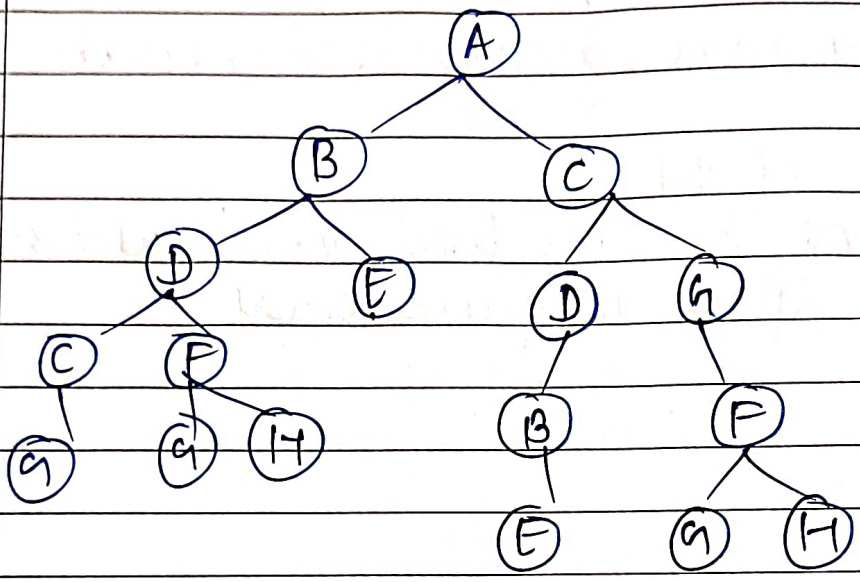
- BFS cannot be effectively used unless the search space is quite small

PAGE

Classmate

DATE

DFS (Depth first search)



Advantage :-

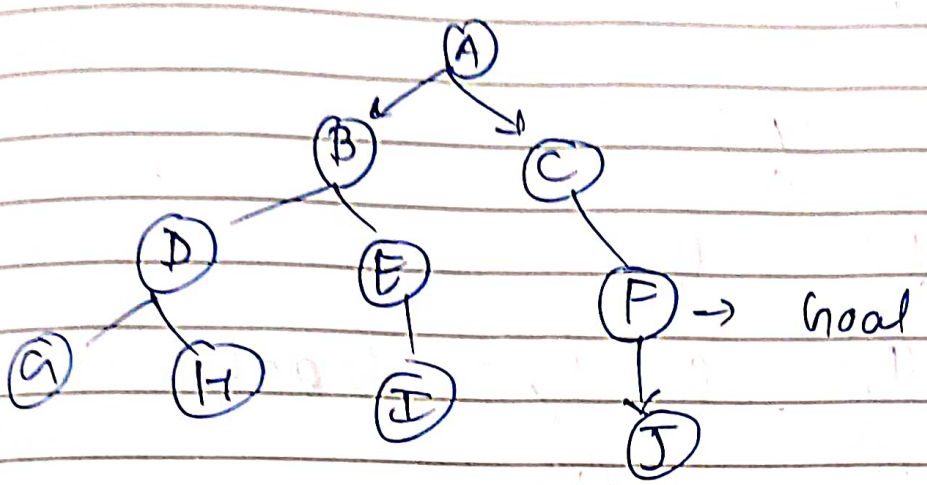
- consums less memory space since only the nodes on the current path are stored.

Disadvantage :-

- may find sub optimal solution.

DATE

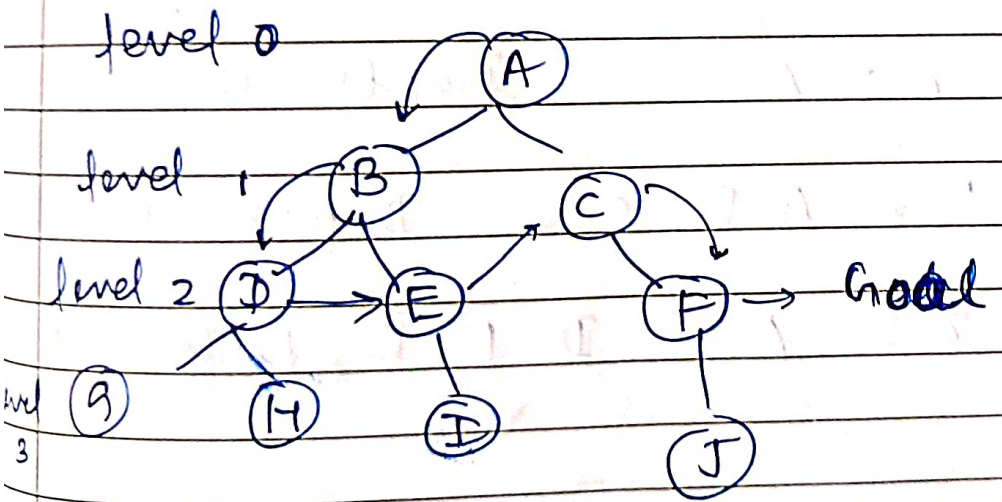
Depth limited search



DFS : left of A

A → B → D → G → H → E → I → C → F

Depth limit

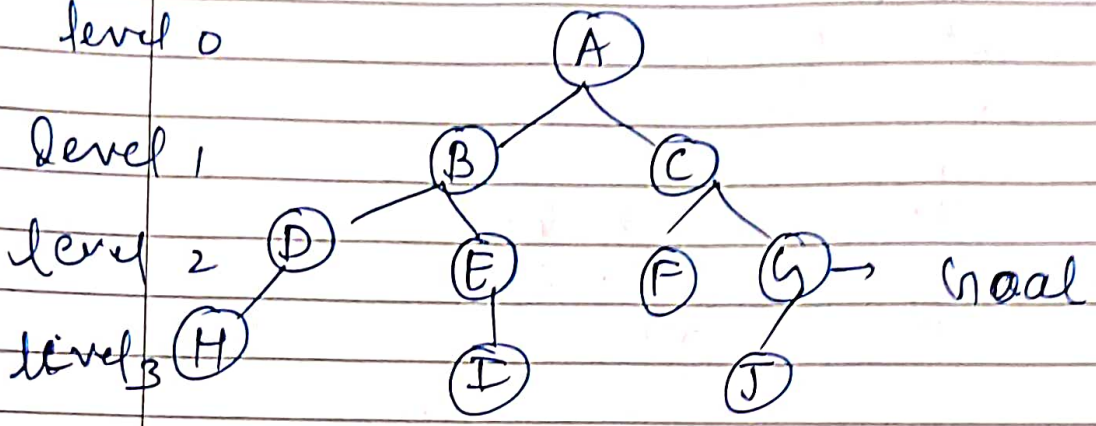


Depth limit 2

DLS : A → B → D → E → C → F

DATE

Iterative Deepening Depth first search



DFS :- A → B → D → H → E → I → C → F → G

iteration 0 : A Depth 0

iteration 1 : A B C Depth 1

iteration 2 : A B D E C F G Depth 2

DATE

Bidirectional search algorithm

Runs two simultaneous searches one from initial called forward search and one from goal node called backward search

It replaces the single search graph with two small subgraph in which one starts the search from initial vertex and other start from goal vertex.

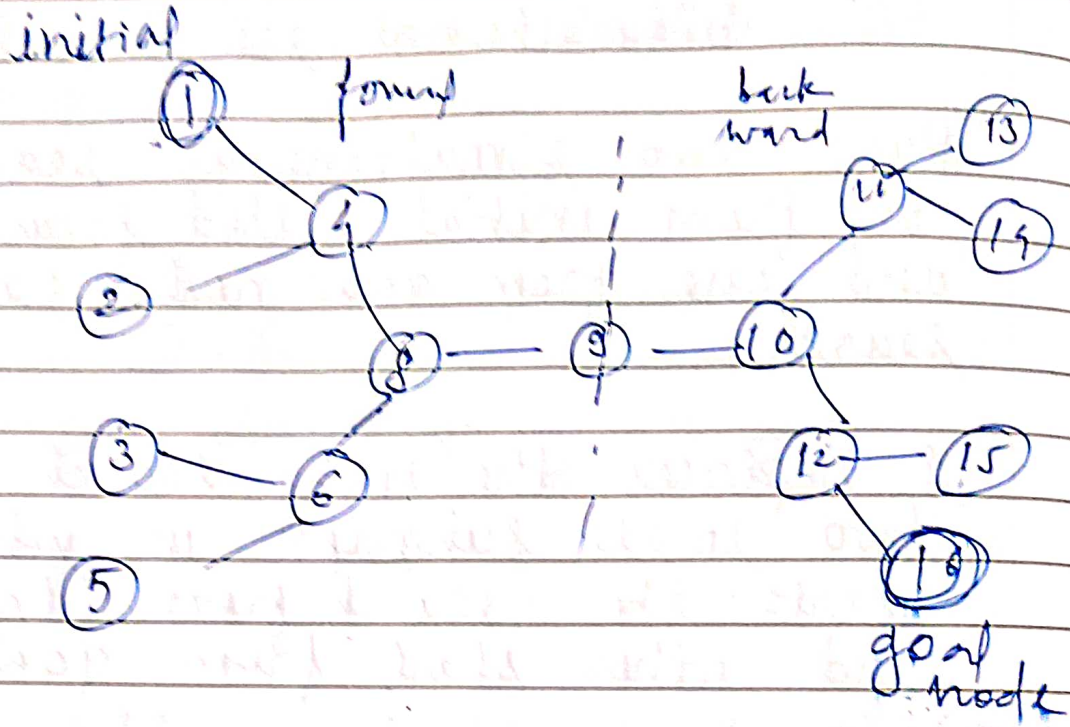
Search stops when these two graphs intersect each other.

BS can use search techniques such as BFS, DFS, DLS etc.

Advantages:
It is fast
requires less memory.

Disadvantages:
implementation is difficult.
should know goal state in advance.

DATE

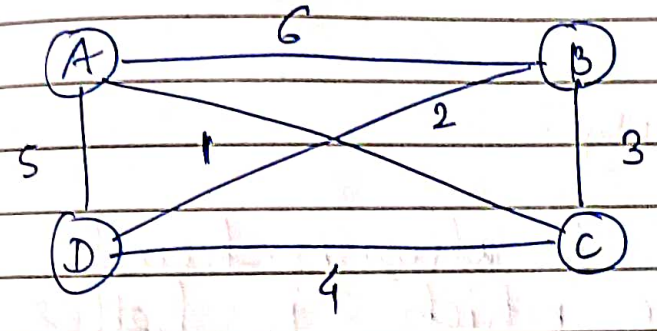


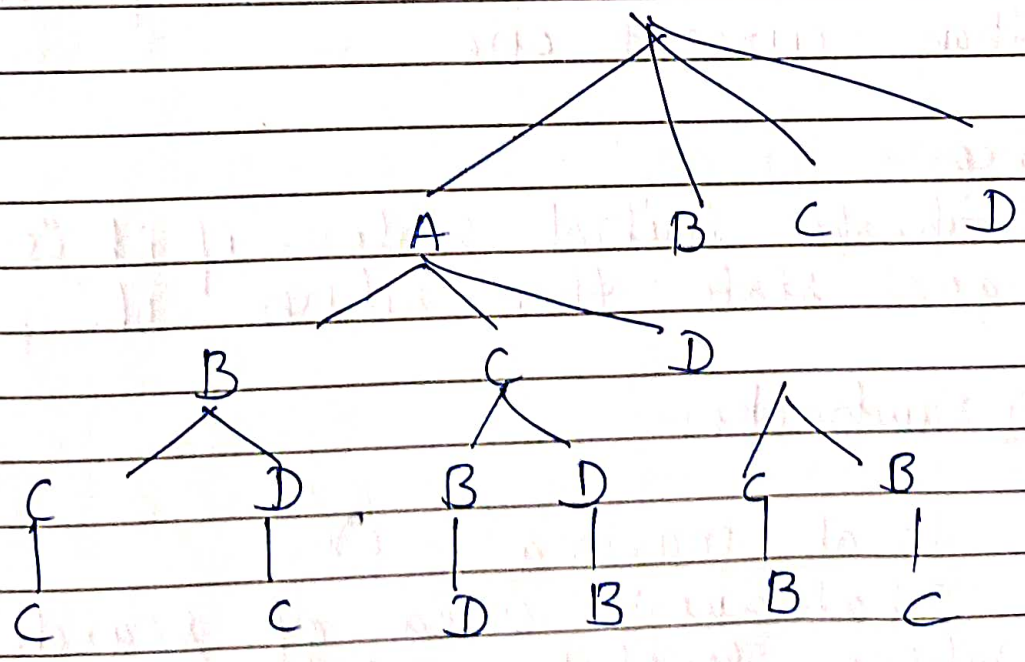
Comparision :-

DATE

Generate and test Algorithm

1. Generate all the possible solutions
2. Select one solution among the possible solutions
3. If solution has been found and acceptable





PAGE

DATE

search for	Path	length
1	ABCD	19
2	ABDC	18
3	ACBD	12
4	ACDB	13
5	ADBC	16

Hill climbing.

The basic idea is to always head toward a state which is better than current one

Steps:-
Evaluate initial state. if it is goal state then return it. , other

Drawbacks:-

local maxima

Plateaus:- area of search space where evaluation function is flat

Ridge

DATE

Example :

start

A −1
D +1
C +1
B −1

(0)

goal

D +
C +1
B +1
A +1

(4)

local heuristic

+1 for each block that is resting on thing it is supposed to be resting on

−1 for each block that is resting on wrong thing

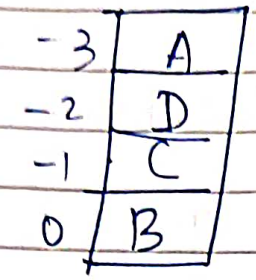
A
D
C
B

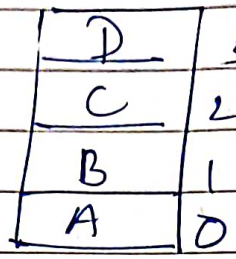
⇒ +1 D
+1 C
−1 B A +1

(2)

⇒ +1 C D −1
−1 B A +1

DATE





-6

6

-2 D
-1 C
0 B A 0

-3

-1 C D -1
0 B A 0

-2

-1 C
0 B A D
0 0

A B C A D

B
A D C

A
B
C
D

DATE

start

A D 3
B C 2
C B 1
D A 0

goal

A -3
D -2
C -1
B 0

-2 D
-1 C
0 B

A
0

-1 C
0 B

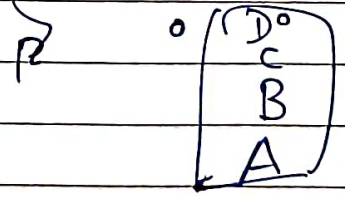
D -1
A 0

C
B

A

D

C B A D
0 0 0



DATE

Diffrence between Hill climbing and Steepest hill climbing method

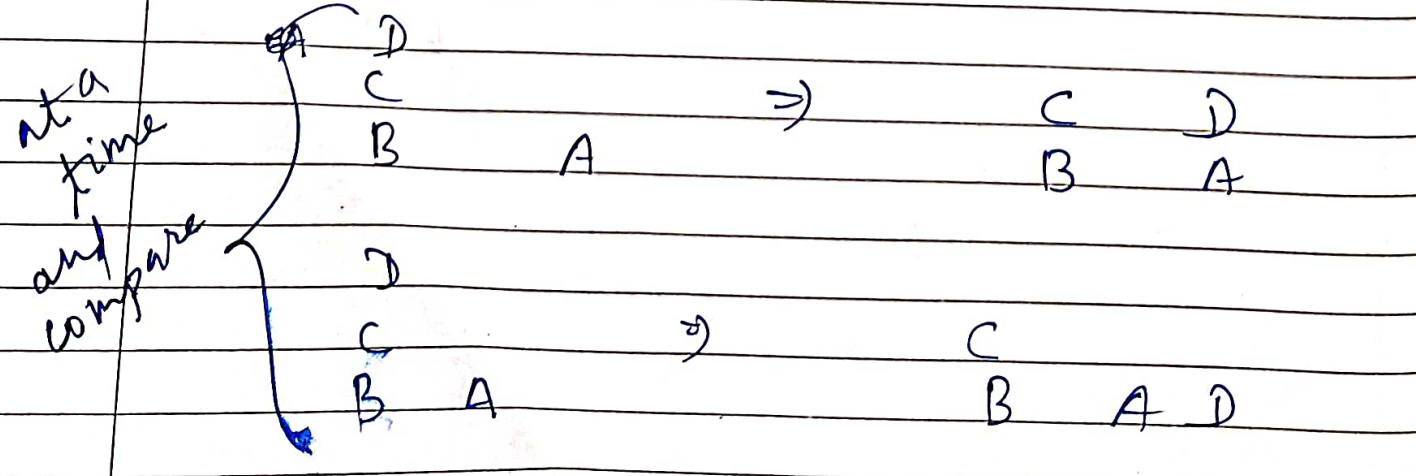
Basic hill climbing applies one operator and gets new state.

if it is better that becomes current state

where as in steepest hill climbing we consider all the moves from current State.

Set.

Selects best one



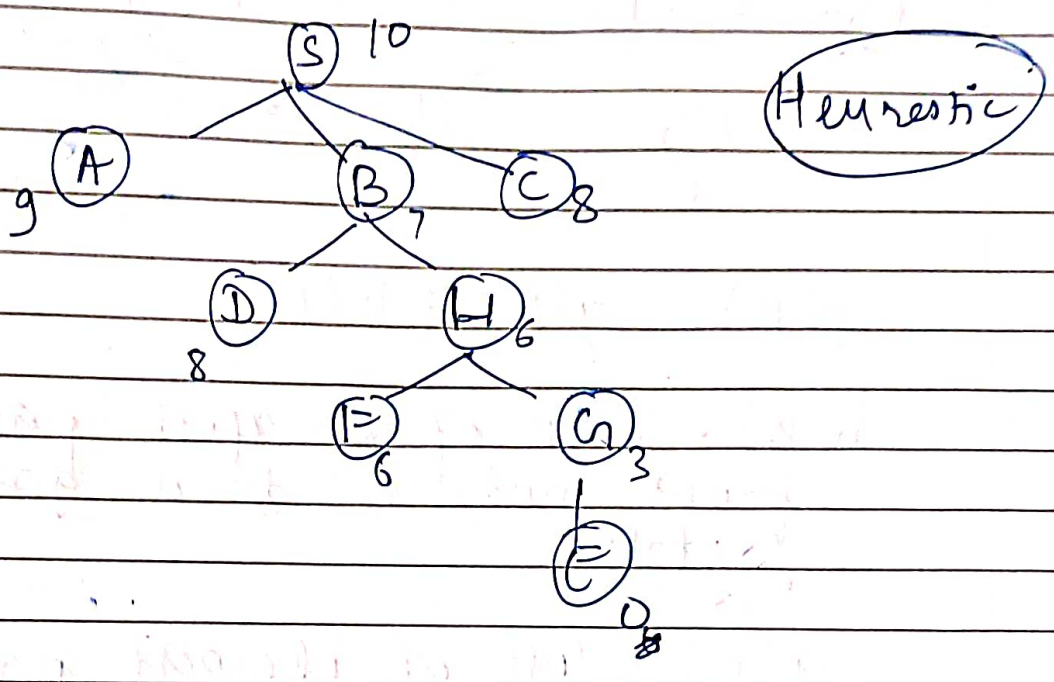
classmate

PAGE

DATE

Best first search

- create 2 empty list OPEN CLOSED



Open

Node	H(n)
S	10
~~A~~	~~9~~ B 7
~~B~~	~~7~~ C 8
~~C~~	~~8~~ A 9

closed

Node	H(n)
S	10
B	7

Open

Node	H(n)
H	6
C	8
D	8
A	9

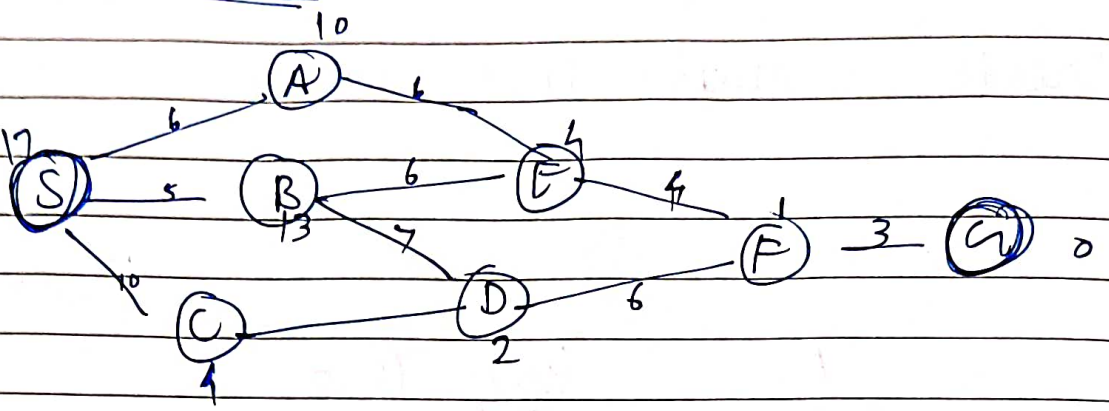
closed

S	10
B	7
H	

S → B → H → G → E

DATE

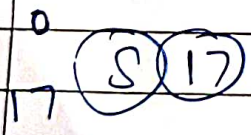
A* Star



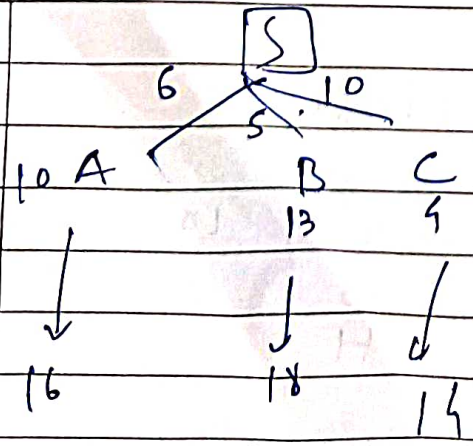
$$f(n) = g(n) + h(n)$$

$h(n)$ = cost of cheapest path from node n to a goal state

$g(n)$ = cost of cheapest path from initial state to node n



Queue
S



Queue
S C
S A
S B

DATE

A* 8 puzzle problem

2	8	3
1	6	4
7		5

initial state

1	2	3
8		4
7	6	5

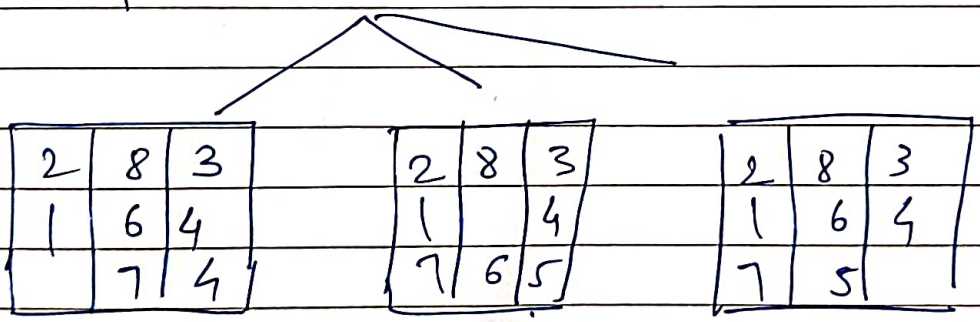
final state

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

$g(n) \Rightarrow$ depth of node

$h(n) \Rightarrow$ numbers of misplaced tiles

$g = 0$, $h = 4$ $\quad f = 0 + 4 = 4$



2	8	3
1	6	4
	7	4

$g = 1$, $h = 5$, $f = 1 + 5 = 6$

2	8	3
1		4
7	6	5

$g = 1$, $h = 3$, $f = 1 + 3 = 4$

2	8	3
1	6	4
7	5	

$g = 1$, $h = 5$, $f = 1 + 5 = 6$

select minimum

DATE

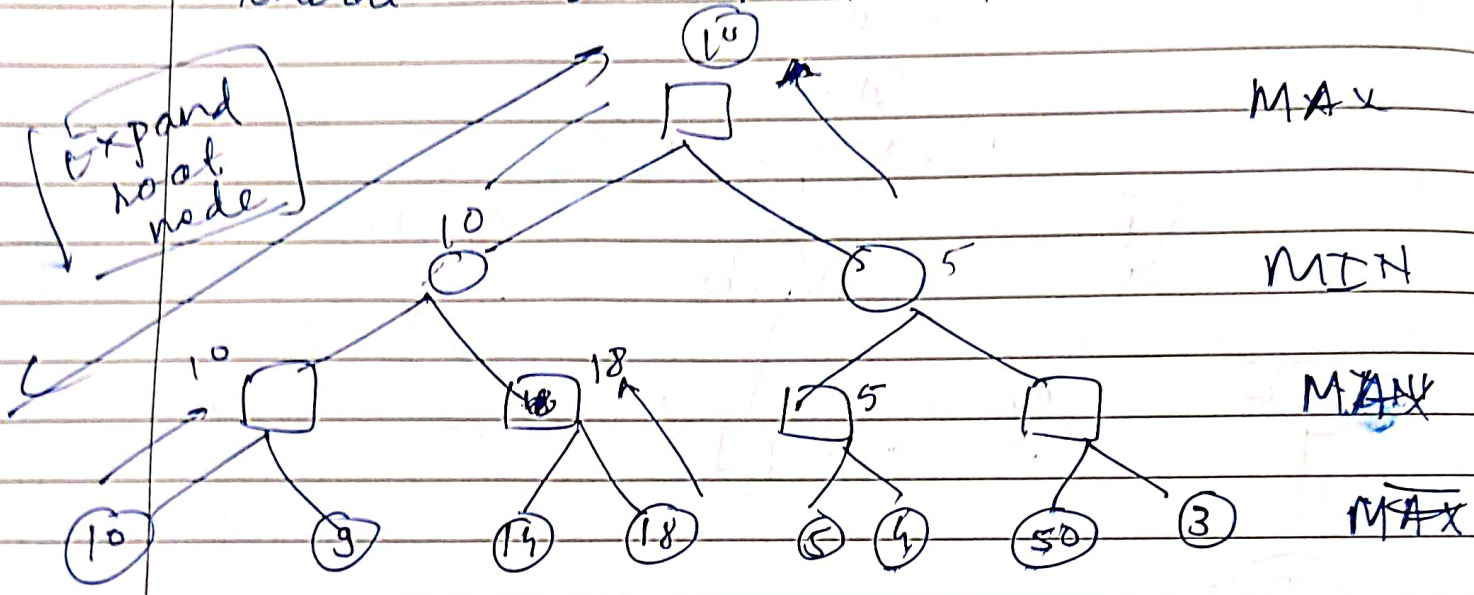
$f(O,I) = 1+1 = 2$

(5) will be replaced by 2

6 will be replaced by 3

Mini Max Algorithm

Generate the whole tree to leaves



Select the path to which 10 is assigned

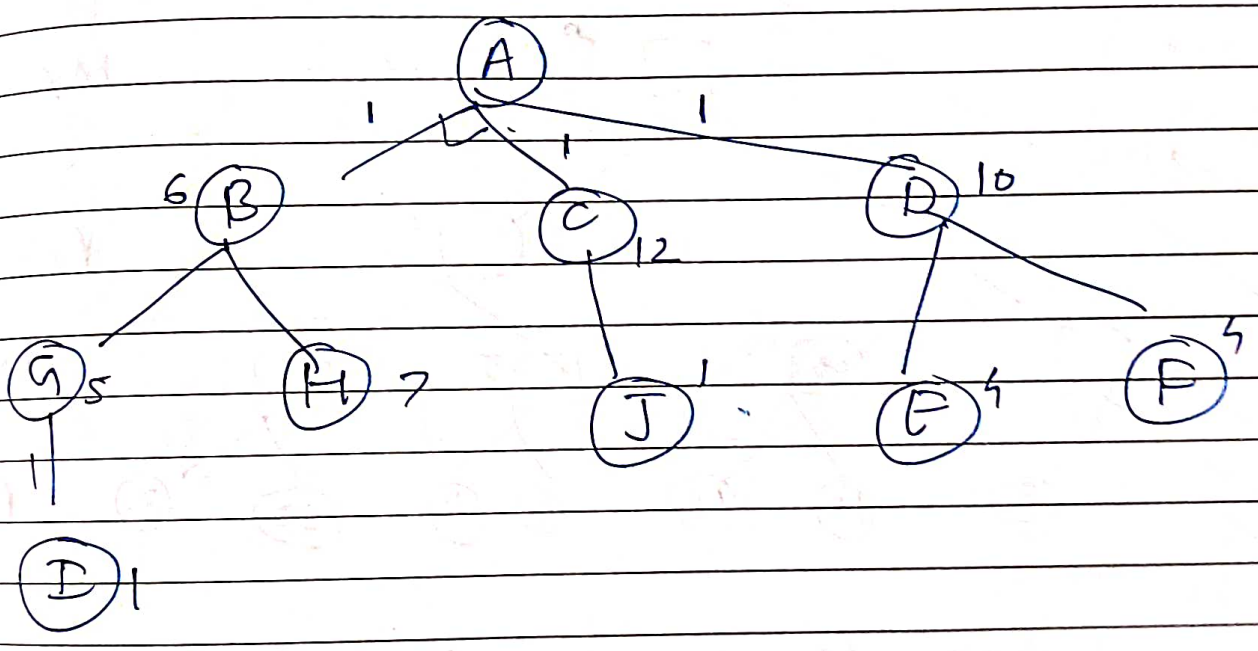
DATE

AO* Search

In AO* we devide ~~small~~ problems into small problems

and Use concept of AND OR graphs

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



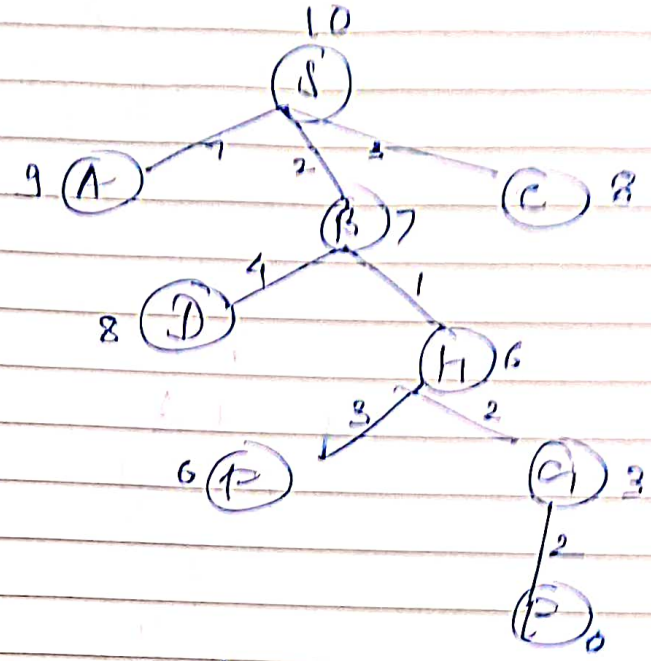
$f(AD) = 10 + 1 = 11$

$f(ABC) = 12 + 8 + 1 + 1 = 20$

$f(DEF) = 4 + 4 + 1 + 1 + 1 = 10$

DATE

A* Search



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

$f(B) = 2 + 7$

$= 9$

Open

Node	g(n)	h(n)	f(n)
→ S	0	10	10
A	7	9	16
B	2	7	9
C	3	8	11

Closed

Node	parent
S	
B	S
H	B

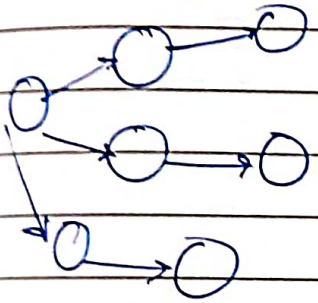
$f(h) = 3 + 6$

$= 9$

D	6	8	14
H	3	6	9

DATE

Uninformed search / Brut force	Informed search
search without information	search with information
No knowledge	Use knowledge
time consuming	quick solution
more complexity (n-1)!	less complexity
	Heuristic



BFS (Breadth first search)

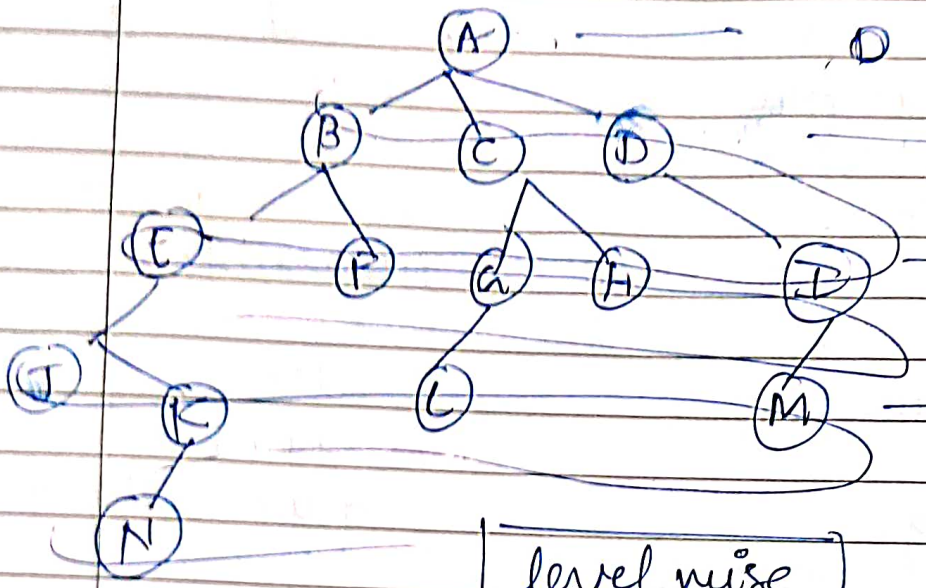
→ uninformed search technique → optimal

→ FIFO (queue)

→

fringe → first node

DATE



d = 1

d = 2

d = 3

d = 4

level wise

A

B C D

C D E F

D E F G H

Time complexity ~~O(V+E)~~

V :- ~~no. of nodes~~

E :- $O(b^d)$

b :- branch factor

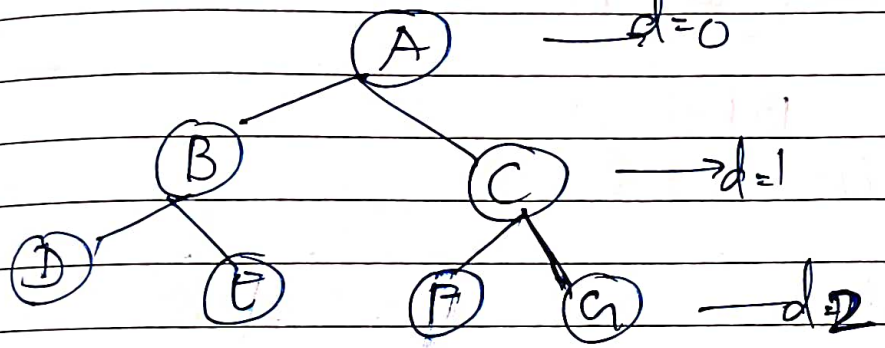
d depth

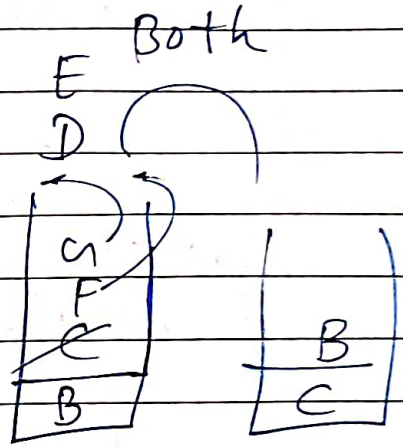
DATE

Depth first search

Uninformed search → Non optimal
stack (LIFO)

Deepest Node
Incomplete.





⇒ A C G F B E D

time complexity → $O(b^d)$
branching factor
depth

DATE

Bidirectional search

Simultaneous search from an initial node to goal and backward ~~to~~ from ~~initial to go~~ goal to initial

$$2\left(b^{d/2}\right)$$

Complete in BFS
incomplete in DFS

Optimal

~~8 puzzle~~ using ~~non heuristic~~

~~BFS~~
~~$O(b^d)$~~