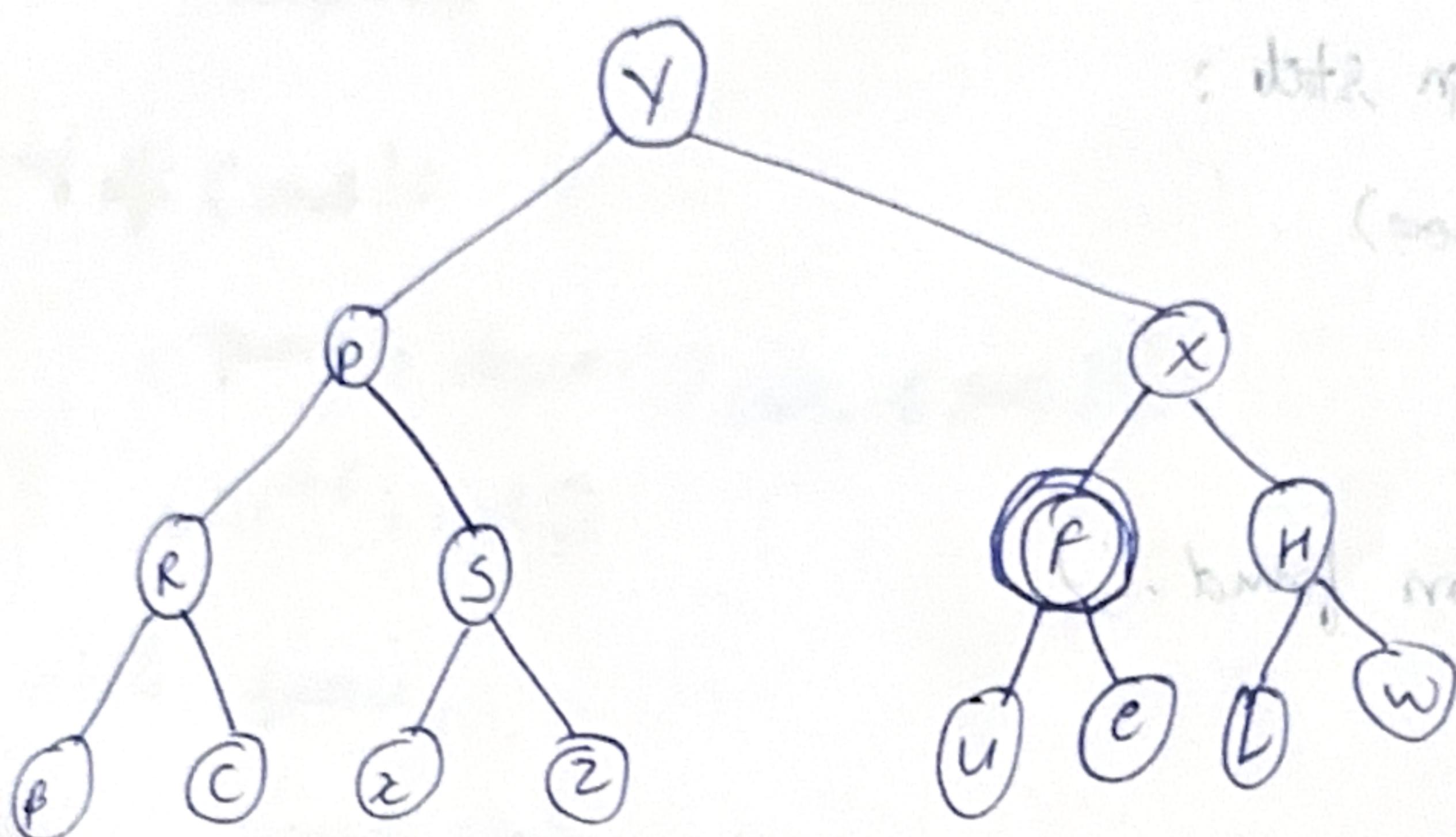


① Iterative depth first search



⇒ Iterative DFS is combination of both DFS & BFS.

Call the depth ~~search~~ search function from range $(0, \text{maxsize})$.

goal = given by user input.

func IDDFS (Graph, limit, start)

for depth=0 to limit:

~~for i in range depth~~

result = DFS (start, depth), limit):

if result:

return result

else

return : null

def DFS (root, limit, depth):

if root = goal:

return : root

if depth == limit+1: return

for Child is root

DFS (Child, level, depth)



Goal state

2	8	1
4	3	
7	6	5

Initial state

1	2	3
8		4
7	6	5

manhattan cost

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

$$h(n) = 2+1+1+1+2+1$$

1	.	3
8	2	4
7	6	5

$$g(n) = 1$$

$$h(n) = 8$$

$$f(n) = 9$$

1	2	3
8		4
7	6	5

$$h(n) = 1+1+2+1+1$$

$$f(n) = 7$$

1	2	3
8		4
7	6	5

$$h(n) = 2+1+1+2+$$

$$1+1+1$$

$$f(n) = 7$$

1	2	3
8	6	4
7		5

$$f(n) = 9$$

$$g(n) = 2$$

.	2	1
1	8	4
7	6	5

$$h(n) = 1+1+1+2+$$

$$1+1$$

$$g(n) = 9$$

1	2	3
8		4
7	6	5

$$h(n) = 2+1+2+$$

$$1+1$$

$$g(n) = 9$$

1	2	3
7	8	4
6	5	

$$h(n) = 2+1+1+$$

$$1+1+1$$

$$g(n) = 9$$

$g(u)=9$

$g(u)=9 \quad g(u)=9$

1	2	3
8	4	
7	6	5

1	2	
8	4	3
7	6	5

1	2	3
8		4
7	6	5

$h(u) = 5$

$g(u) = 9$

$g(u) = 3$

1		2
8	4	3
7	6	5

$$h(u) = 2 + 2 + 2 \\ = 6$$

$g(u) = 9$

1	2	3
8	4	
7	6	5

$$h(u) = 2 + 1 + 1 + 2 \\ = 6$$

$g(u) = 9$

Algorithm - Draft 10/10

def A* (currentstate, goalstate)

Cost = 0

If (current state != visited):

States[] = generate states (current state)

for ~~for~~ i in states:

fun = Cost + manhattanDistance (state)

min = min (fun)

visited.append (state)

A* (state, goalstate)

Output

IDDFS:

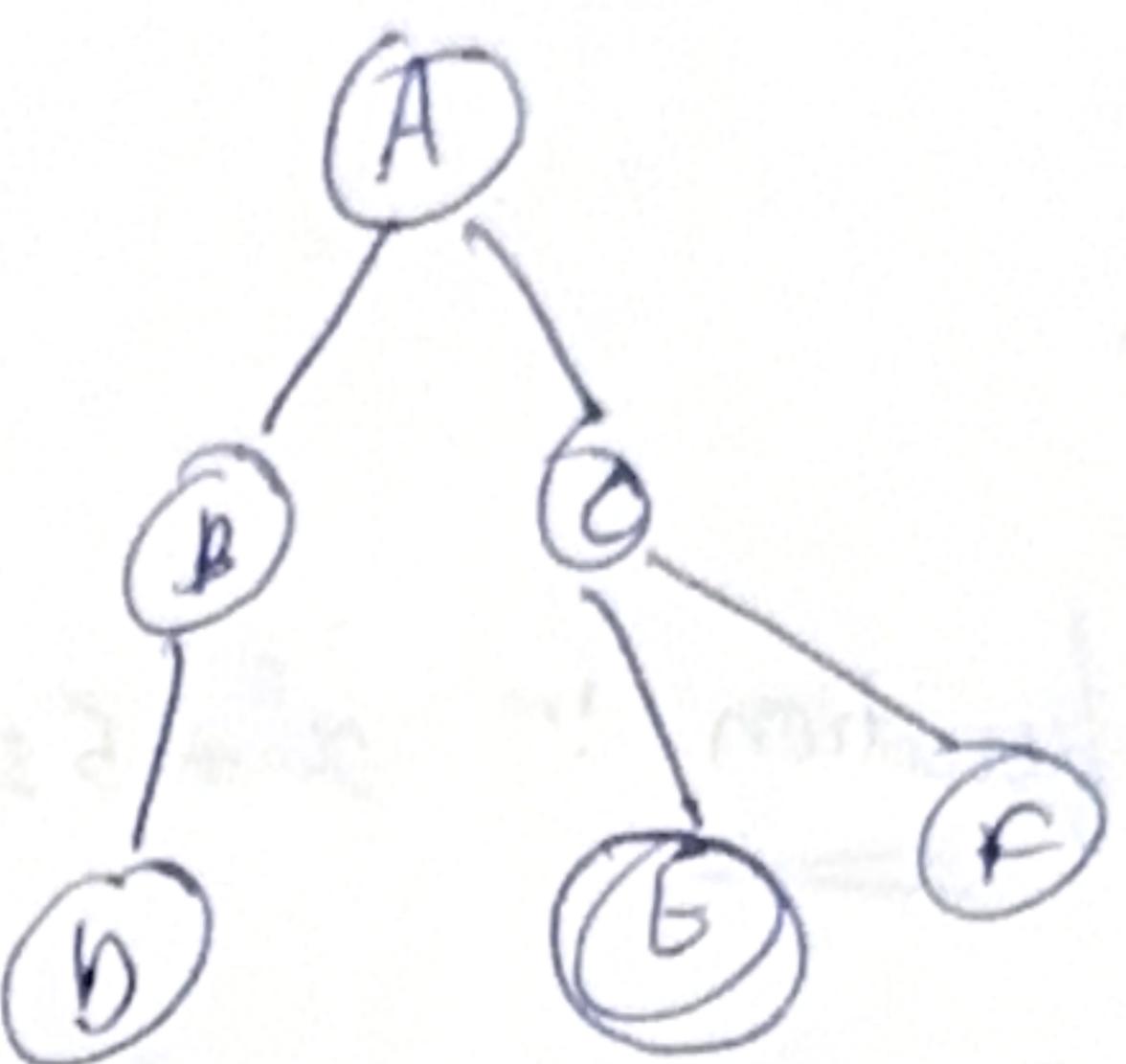
Iteration 1:

$A \rightarrow B \rightarrow C \rightarrow$

Iteration 2:

$A \rightarrow B \rightarrow D \rightarrow E \rightarrow$

Target node E found



$\Rightarrow A^*$

Output

Start = $(1, 2, 3, 4, 5, 6, 0, 7, 8)$ \Rightarrow goal - letters = goal

goal = $(1, 2, 3, 4, 5, 6, 7, 8, 0)$

$\Rightarrow (1, 2, 3, 4, 5, 6, 0, 7, 8)$ \Rightarrow goal at 1 $\rightarrow 1$ so!

$(1, 2, 3, 4, 5, 6, 7, 0, 8)$ \Rightarrow goal - letters = goal - new

$(1, 2, 3, 4, 5, 6, 7, 8, 0)$ \Rightarrow goal - letters < goal - new

$(1, 2, 3, 4, 5, 6, 7, 8, 0)$ \Rightarrow goal - letters < goal - new

$(1, 0) \text{ valued} < (\text{goal}, \text{tree-new}, \text{tree-new}) \text{ value}$

tree-new = tree-new

tree-new > tree-old ?

tree-new = tree-old

tree-new > tree-old

tree-new < tree-old

$(\text{tree-old}, \text{tree-new}) \text{ value}$