Homework01

Tuesday, September 19, 2023 10:10 PM

Kamal Giri, Homework #1

Qno1

Solution

Let's try to understand the algorithm:

Our algorithm is just a basic search algorithm.

(a) If a FIFO queue is is used then our algorithm will act like a breath first search (BFS). where let d be the depth of the tree and each node of the tree has at most b children then the total number of nodes that can be visited will be

$$1 + b + b^2 + \dots + b^d = O(b^d)$$

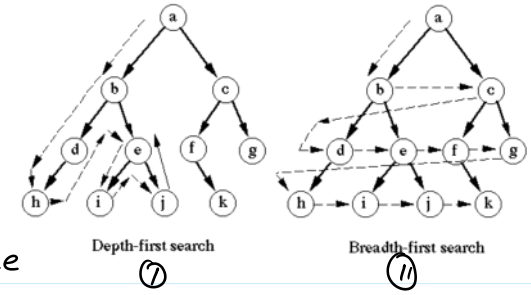
which is the worst time complexity.

Hence, (b^d) will be the largest number of nodes simultaneously stored in the queue as we are

i.e $b = 8$ & $d = 3$

$= (8^3)$

$= 512$ nodes



Depth-first search (7) Breadth-first search (11)

As we saw in our picture (11), we have to keep all our node in queue to search k. from a.

(b) If we are using LIFO, then our search method is Depth First Search.

In case of depth first search, we are using LIFO, so our frointier/ stack has to store only the single path from the root to leaf node as other remaining unvisited siblings. for each node in the path.

Hence, the max amount of node stored will be

Hence, the max amount of node stored will be

bd where b = branching factor

d = depth of the tree

Hence, we are given, $b=8$, $d=3$

so, largest number of nodes simultaneously stored = 24.

(c) In any algorithm, if we don't find the our goal state then we might have to store and pop all the nodes we have generated which will be

$$N(\text{worst}) = db + (d-1)b^2 + (d-2)b^3$$

$$= 3\times 20 + 2\times 64 + 1\cdot 8^3$$

$$= 700$$

Qno 2

Given,

each step cost (ε) = $\frac{1}{7}$

each node has b children (b) = 2

Cost of optimal path (C^*) = 10

Now,

Since, this is uniform cost search,

we know that worst-case asymptotic running time

is $O(b^{\lfloor 1 + \lfloor C^*/\varepsilon \rfloor \rfloor})$

Solving

$$b^{\lfloor 1 + \lfloor C^*/\varepsilon \rfloor \rfloor}$$

$$= 2^{\lfloor 1 + \lfloor 10/\frac{1}{7} \rfloor \rfloor}$$

$$= 2^{\lfloor 1 + \lfloor 70 \rfloor \rfloor}$$

$\Rightarrow 2^n (1+70)$

$\Rightarrow 2.361 \times 10^{21}$

Qno3

Given

6	9	8
7	1	3
2	5	4

initial state

(a) (i) initial state is given above

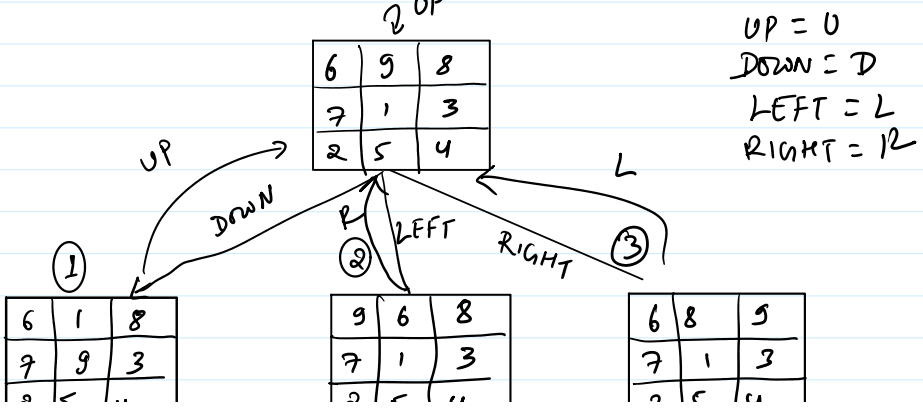
(ii) other states are obtained by swaping 9 with other numbers until we reach goal state

(iii) actions are UP, DOWN, LEFT & RIGHT

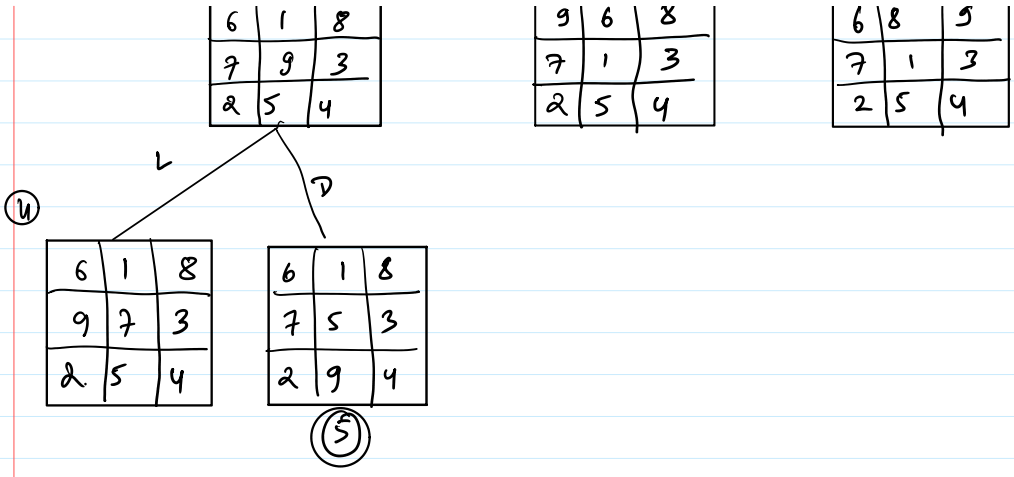
(iv) Goal condition is achieved when all 3 number in all rows, clums, diagonals give sum of 15.

(b) This is graph.

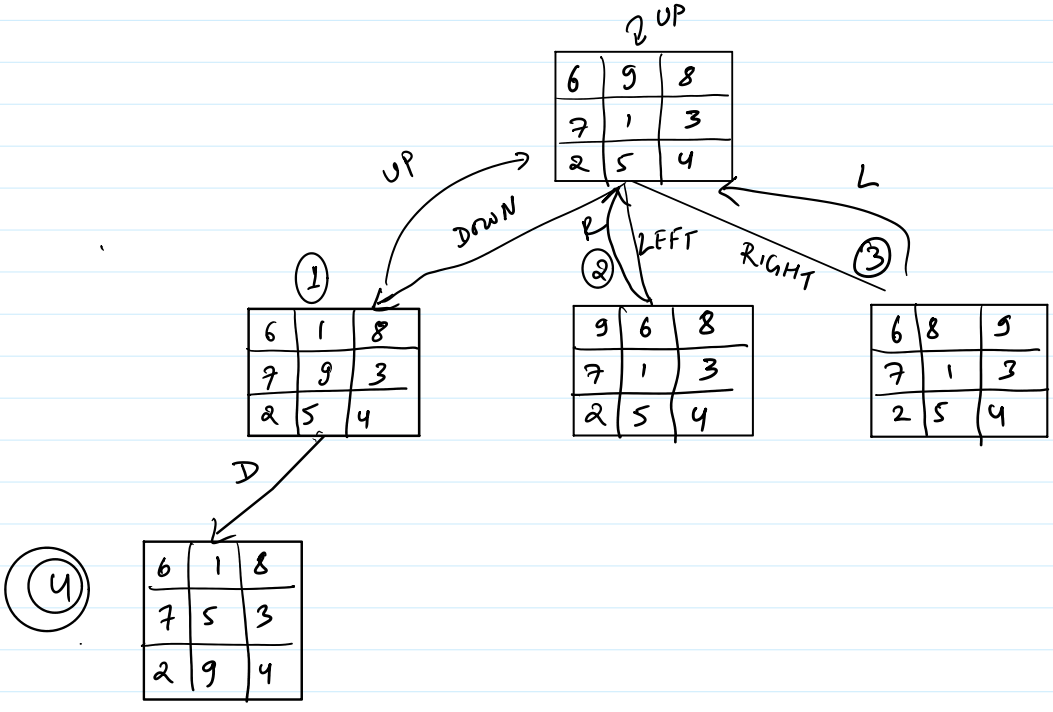
(c) portion of graph generated by Breath first search



UP = U
DOWN = D
LEFT = L
RIGHT = R



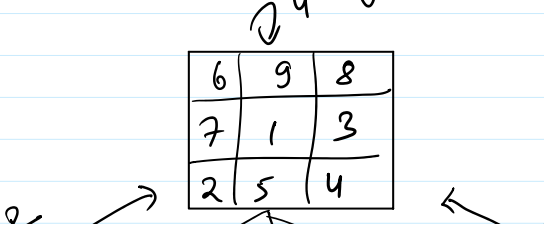
(d) portion of the Graph generated by Depth-First search.

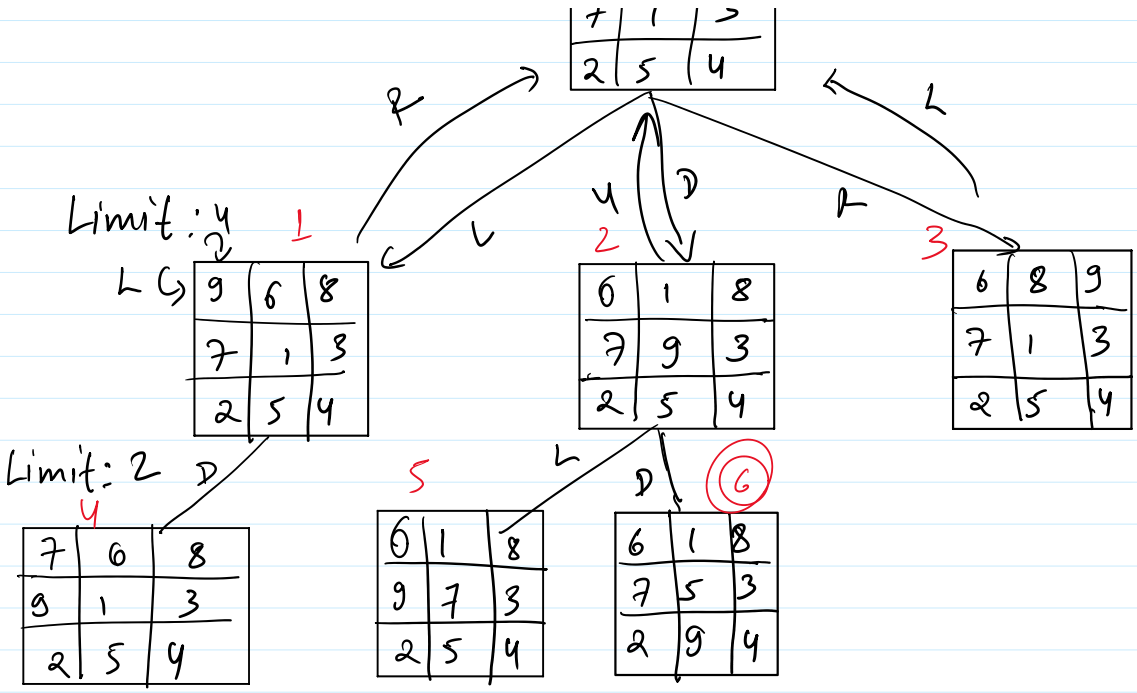


(e) portion of the state generated by Iterative Depth search.

The iterative depth search repeatdely applies depth limited search with increasing limit. until the goal node is not found.

Limit : 0




Limit: 4
Limit: 2
1
2
3
5
6
R
L
U
D