Name : Md Raihanul Islam Bhuiyan

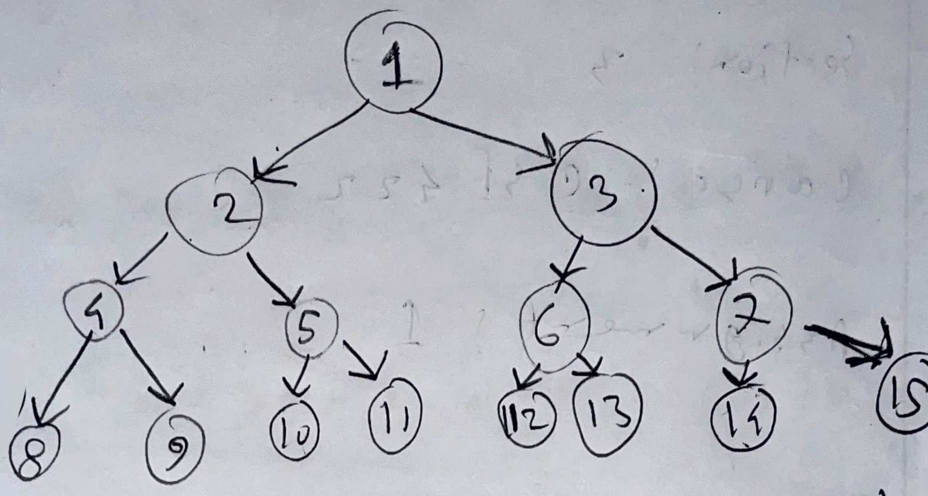
Id : 20101239

Section : 3

Course : CSE 422

Assignment : 1

Ans to the Q: N: 1



Here, BFS will be suitable to reach both the goals 6 and 10.

If we are looking for goal node 10, the order for the expansion will be like this,

1, 2, 3, 4, 5.

After we expand the node 5, we find the goal node 10 and we can stop

BFS then. We had to expand 5 nodes only.

If we applied a left handed DFS, we would need to expand like following:

1, 2, 4, 8, 9, 5, 10

We would need to expand 7 nodes for this.

So, BFS will be better here.

For the goal node ~~to~~ 6, BFS works better again. The expansion order is,

1, 2, 3.

If we applied DFS, it would be,

1, 3, 7, 15, 14, 6, 13, 12.

So, definitely BFS will be better here also.

Ans to the Q: N: 2

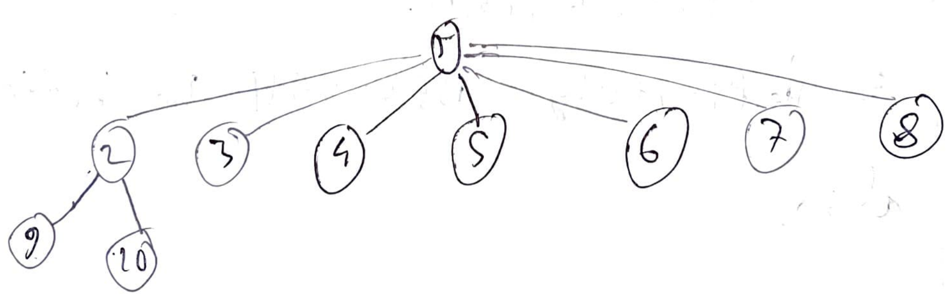
(a)

In iterative deepening search, we apply DFS for each level of the tree.

Time complexity of IDDFS is, $O(b^m)$

Here, b indicates the branching factor

When the value of b is high, then ~~the~~ DFS is better. for example,



If our goal node is 9 here, then reaching the goal node will be much faster with DFS.

So, when the branching factor is higher, DFS outperforms Iterative Deepening Search.

(b)

Yes, BFS is complete even if zero-step costs are allowed.

An algorithm is complete when it can ensure that the goal nodes can be found in every cases whether it is optimal or not. In Breadth-first Search, if the goal states are at a finite depth d, then it can always find the goal. Step costs are irrelevant here.

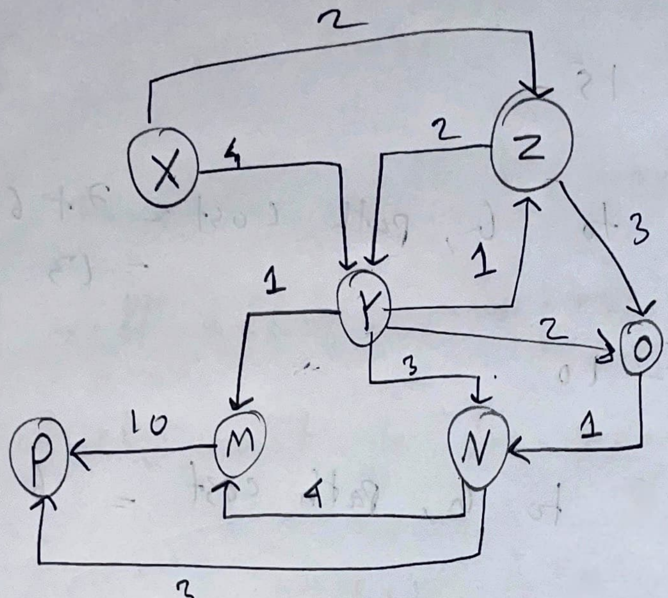
Ans to the Q: N: 3

In Uniform Cost Search, we expand the nodes based on which have the shortest path cost. Every time we append in the queue, we sort the ~~childs~~ children based on the path cost.

On the other hand, plain BFS does not check the path cost at all. Every time we expand a node, we just append the children in the queue. There is no sorting involved.

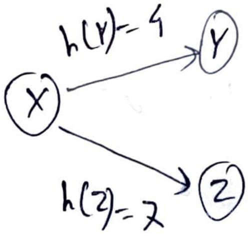
Another difference is, whenever a goal node is generated, we can stop BFS. But in case of UCS, we ~~not~~ need to keep searching till we expand the goal node.

Ans to the Q: N: 4

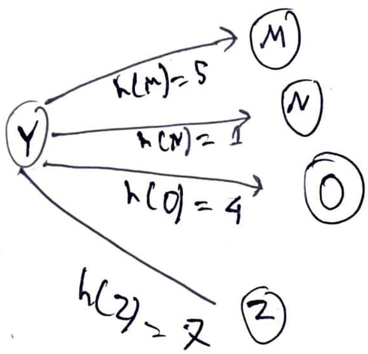


X	7
Y	4
Z	7
M	5
N	1
O	4
P	0

Greedy Best First Search

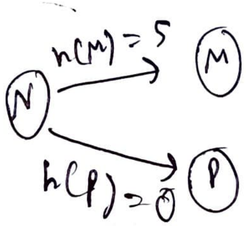


from X, we can go to Y or Z. As h(Y) is lower, we choos Y. So, X → Y.



From, Y, we can go to M, N, O or Z, As h(N) is lower, we choose N. So,

X → Y → N.



$\therefore X \to Y \to N \to P$.

This is the path the Greedy Best First Search will traverse to find the goal node P.

A* Search Algorithm

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

$X >$ $f(n) = 0 + 7 = 7$ ✗

$X > Y$ $f(n) = 4 + 4 = 8$ ✗

$X > Z$ $f(n) = 2 + 7 = 9$ ✗

$X > Y > M$ $f(n) = 4 + 1 + 5 = 10$ ✗

$X > Y > N$ $f(n) = 4 + 3 + 1 = 8$ ✓

$X > Y > O$ $f(n) = 4 + 2 + 4 = 10$ ✗

$X > Y > Z$ $f(n) = 4 + 1 + 7 = 12$ ✗

$x > Z > Y$ $f(n) = 2+2+4 = 8$ ✗ $[X>Y = X>Z>Y]$

$x > Z > O$ $F(n) = 2+3+4 = 9$ ✗

$X > Y > M > P$ $f(n) = 4+1+10+0 = 15$ ✗

$X > Y > N > M$ $f(n) = 4+3+4+5 = 16$ ✗

$X > Y > N > P$ $f(n) = 4+3+3 = 10$ ✗

$X > Y > O > N$ $f(n) = 4+2+1+1 = 8$ ✗

$X > Y > Z > O$ $f(n) = 3+1+3+4 = 12$ ✗

$X > Z > O > N$ $f(n) = 2+3+1+1 = 7$ ✗

$X > Z > O > N > M$ $f(n) = 2+3+1+4+5 = 15$ ✗

$\boxed{X > Z > O > N > P \quad f(n) = 2+3+1+3 = 9}$ ✓

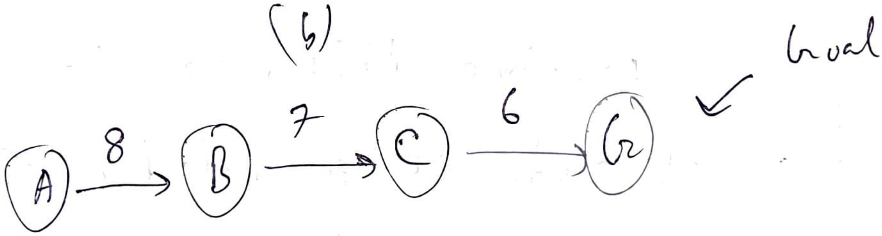
So, if we follow A* search algorithm, the graph will traverse like, X → Z → O → N → P

Ans to the Q: N: 5

(a)

A heurestic will be admissible when it will always underestimate the actual cost to the goal.

(b)



Heurestic Table	
A	15
B	10
C	3
G	0

From A to G, path cost = 8+7+6
= 21

h(A) = 15

from B to G, path cost = 7+6
= 13

h(B) = 10

from C to G, path cost = 6

h(C) = 3

Here, every time the path cost is greater than the heuristic value. So, ~~the~~ heuristic is admissible here.