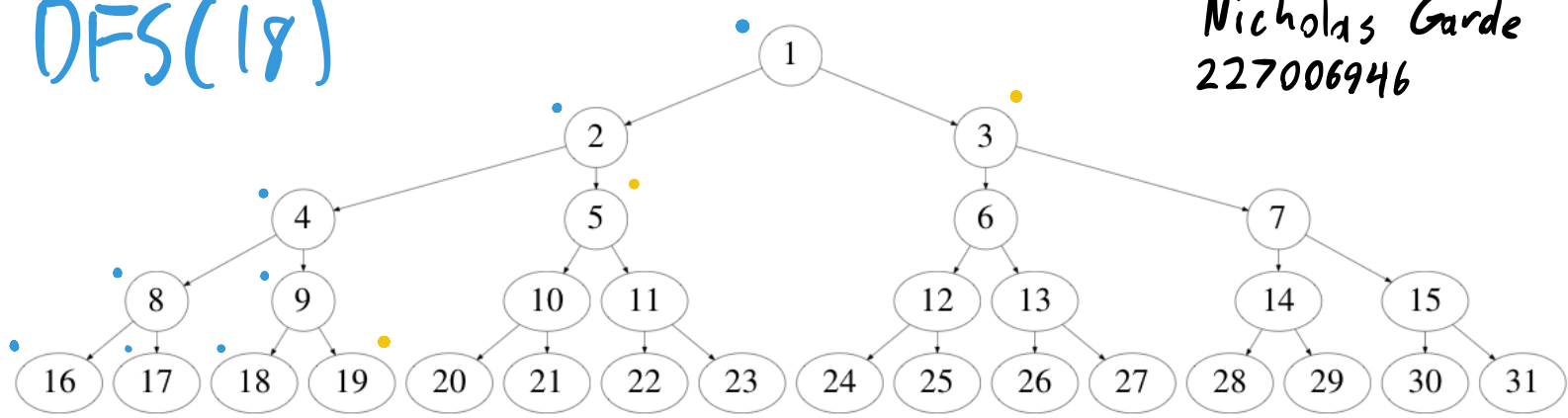


DFS(18)

Nicholas Garde
227006946



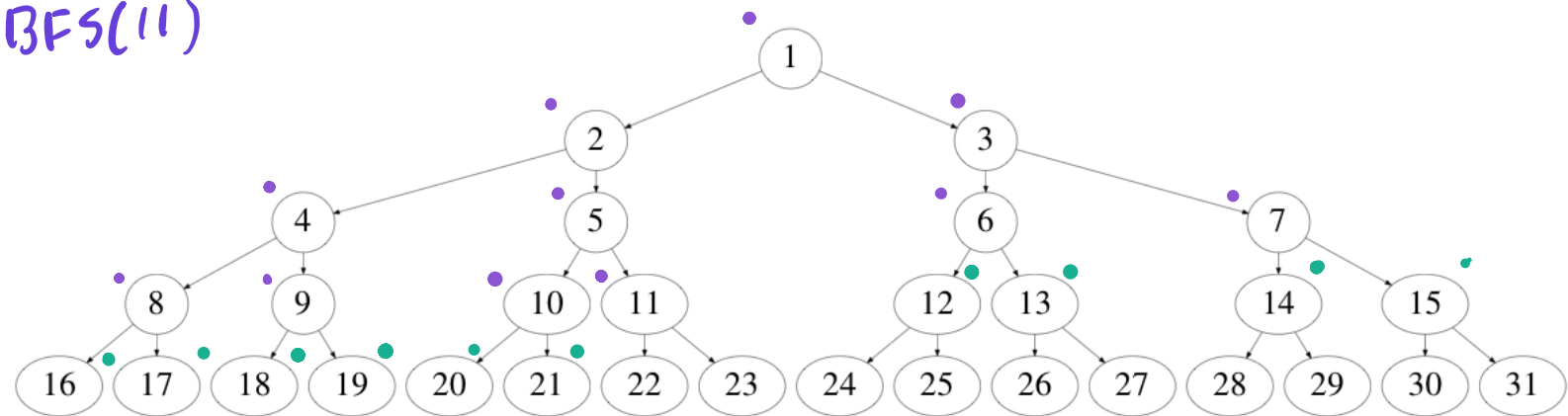
nodes left in queue:

19, 5, 3

nodes visited:

1, 2, 4, 8, 16, 17, 9 before finding (18) goal node

BFS(11)



nodes left in queue:

12, 13, 14, 15, 16, 17, 18, 19, 20, 21

nodes visited:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10 before finding (11)

Question 3 (20 pts): Why is the space complexity of BFS $O(b^{d+1})$, not $O(b^d)$, where b is the branching factor and d is the goal depth?

the queue contains each node at d -depth's b children, this results in b^{d+1} or the number of nodes at $d+1$

Question 4 (20 pts): Can depth limited search become incomplete in the case of the finite search tree above? If so, give an example (use Figure 1). If not, explain why not (use Figure 1).

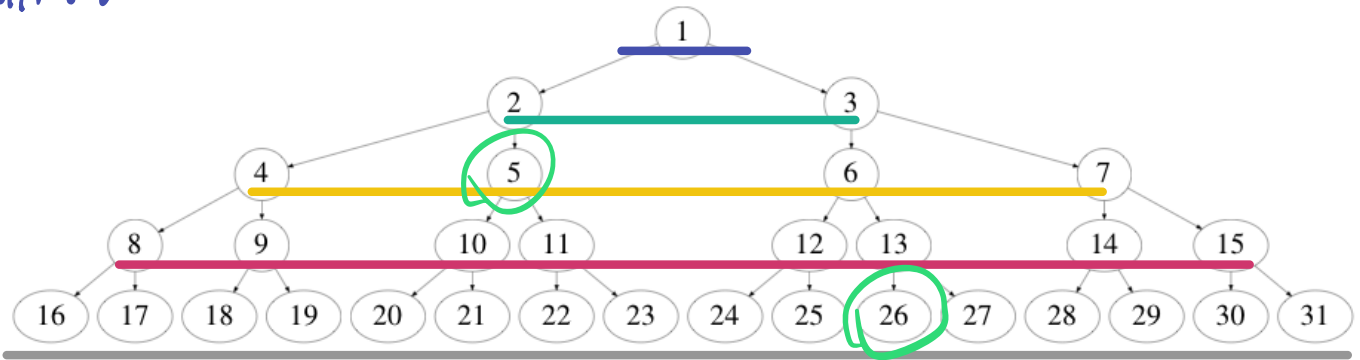
Depth Limit: 0

DL: 1

DL: 2

DL: 3

DL: 4



each of the lines above mark where a depth limit can only reach the nodes above it. only $DL \geq 2$ can find target (5)
only $DL \geq 4$ can find target (26)
if the chosen DL doesn't reach its target, the search is incomplete

Question 5 (20 pts): Consider iterative deepening search. When the goal is (17), how many nodes are visited in total? The count should include all the iterations over the increasing depth limit. Include the visit to (17) in the count.

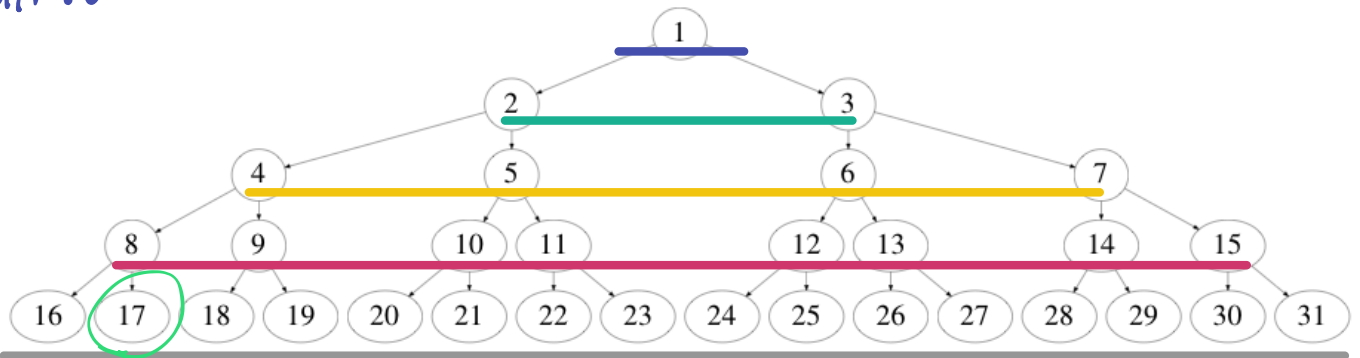
Depth Limit: 0

DL: 1

DL: 2

DL: 3

DL: 4



1; 1, 2, 3; 1, 2, 4, 5, 3, 6, 7; 1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15;
1, 2, 4, 8, 16, 17;

$$1 + 3 + 7 + 15 + 6 = 32$$