Exercise 1

1.

a. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

b. 1, 2, 5, 9, 6, 3, 4, 7, 8, 10

c. ~~1, 2, 3, 4, 5, 6, 3, 7, 8, 9, 6, 3, 7, 10~~

i = 0 → 1

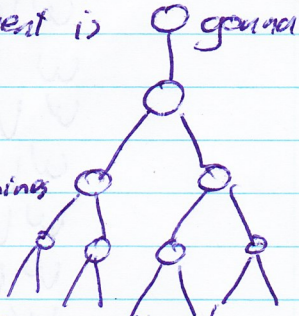
i = 1 → 1, 2, 3, 4

i = 2 → 1, 2, 5, 6, 3, 4, 7, 8

i = 3 → 1, 2, 5, 9, 6, 3, 4, 7, 8, 10

2.

a. The agent is at one point going to go back. If the agent picks north/south or east/west after each other, it will go back. If ~~it~~ there always comes a point where the agent is gonna go back and forth.

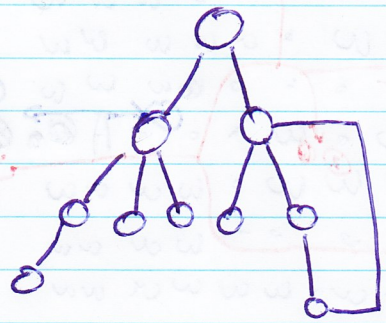


b. Yes, ~~when the~~ at the circle determing with path the agent chooses

c. There should be a statement that the agent can't go back to the last place it came from.

3. IDS vs DFS

a.



When a node in a tree is connected to an earlier node, Depth DFS is going to ~~end~~ get stuck in an infinite loop, where IDS is going to find the other nodes as well.
IDS also finds the ~~sh~~ (more) the shortest solution where DFS finds the first one.

IDS vs BFS

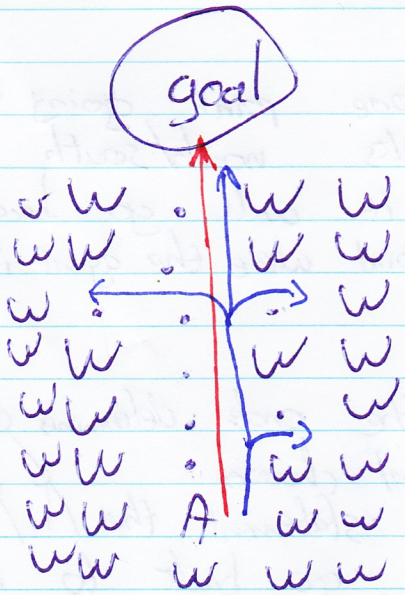
BFS finds the most optimal solution, the fastest but requires a lot more memory because it doesn't finishes the nodes it finds.

a) Running times:

	Time	Complexity Space
BFS:	$O(b^d)$	$O(b^d)$
DFS:	$O(b^m)$	$O(bm)$
IDS:	$O(b^d)$	$O(bd)$

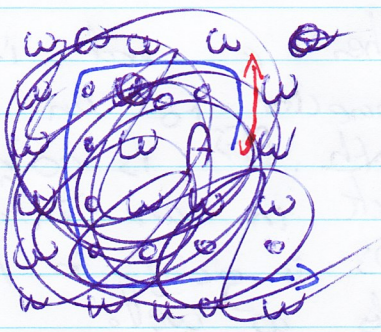
b

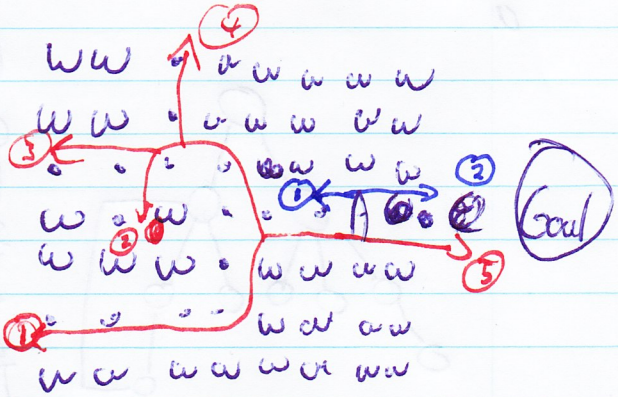
1.



- : DFS
— : IDS

2.





3 DFS has trouble when the goal depth is low and the other branches are more deep then the goal depth.

IDS has trouble if the goal is on a deepe branch because it searches laye for laye.