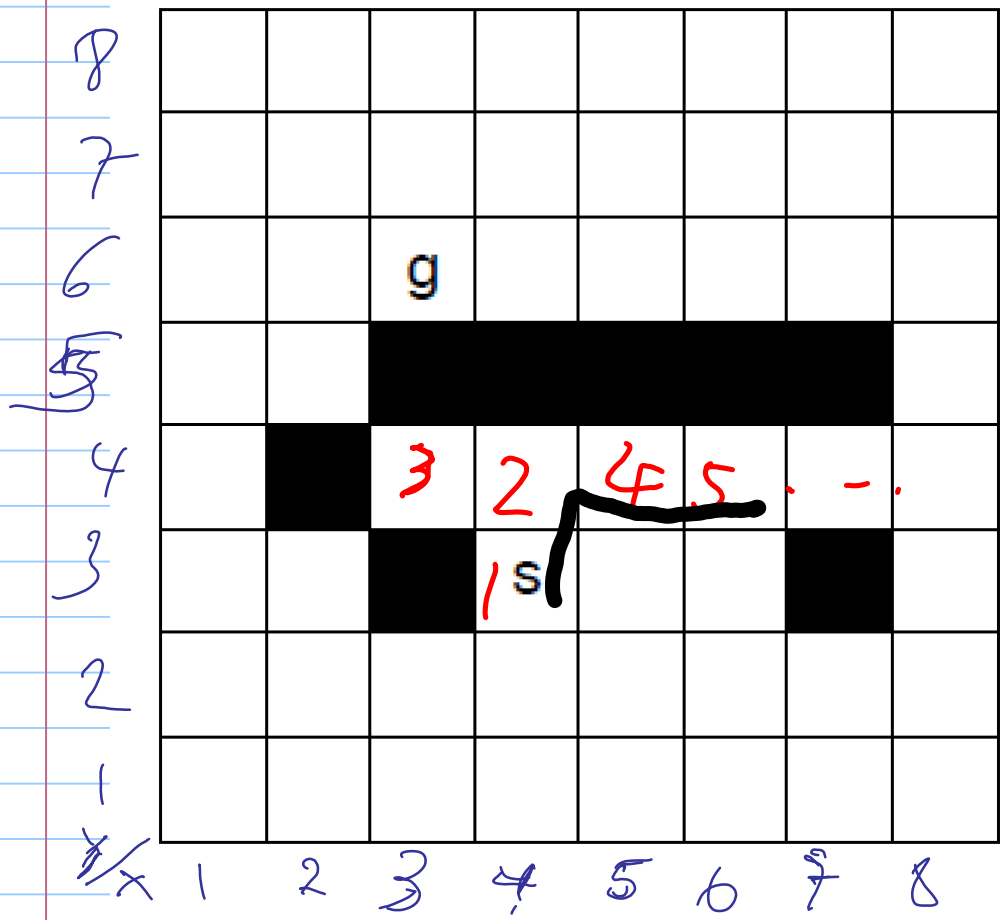
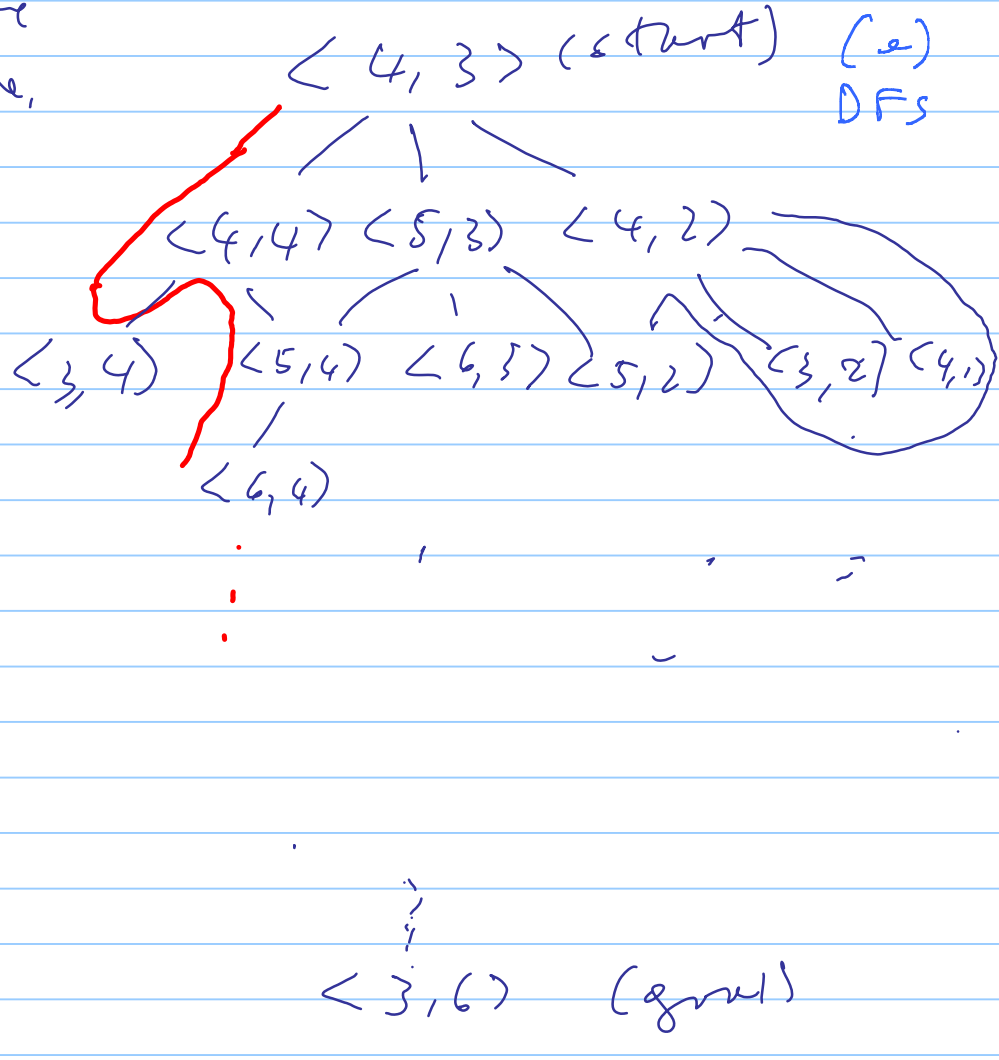
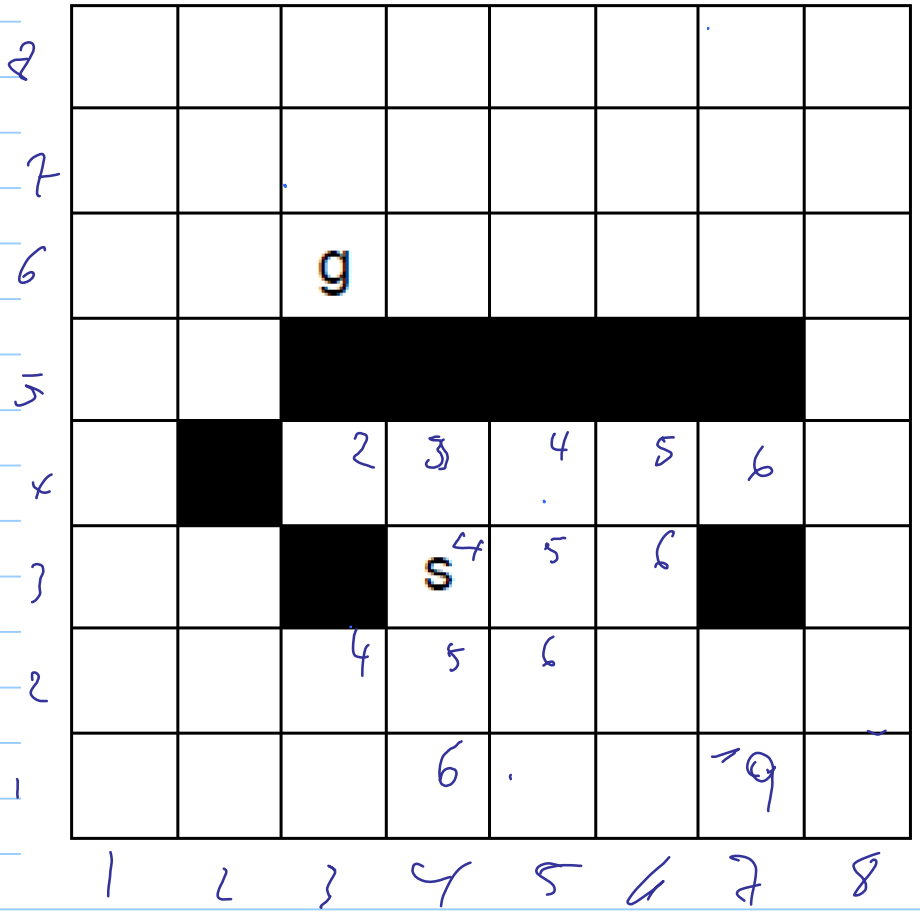
All edge costs are 1. All moves are reversible.





(b) Best-first search. Use two grids: the first one for Manhattan distance



$h(g) = h(\langle 3,6 \rangle) = 0$

$h(\langle x_n, y_n \rangle) = |x_n - x_g| + |y_n - y_g|$

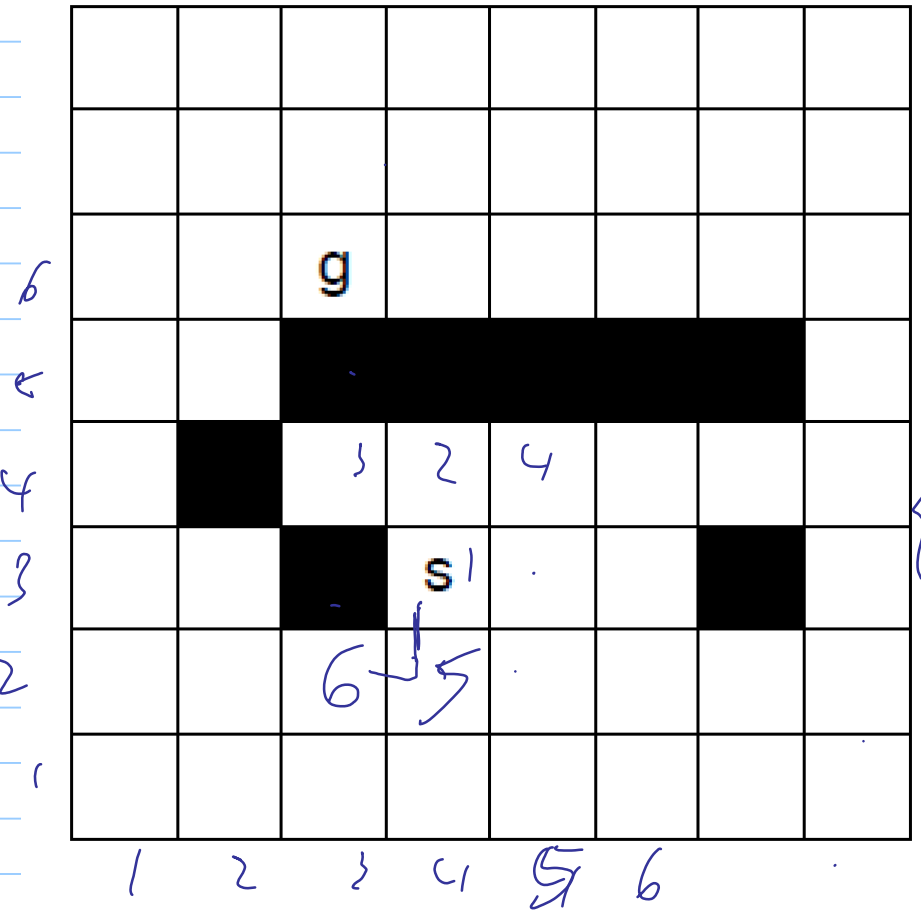
$= h(n)$

$h(s) = h(\langle x_s, y_s \rangle) = h(\langle 4,3 \rangle) =$

$= |4 - x_g| + |3 - y_g| = |4-3| + |3-6|$

$\geq 1 + 3 = 4$

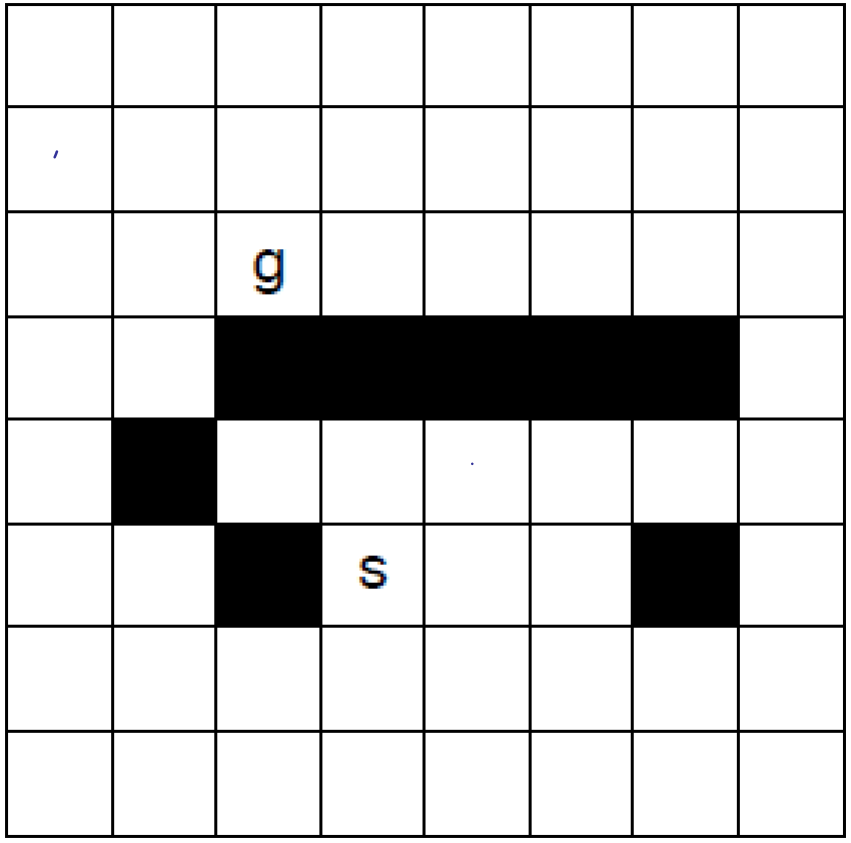
(b) part 2 - Part 5 needs two grids. Here you do the same work as for part (a) but using Best-First Search with Manhattan distance as the h function.



OPEN	CLOSED
(4,3)	
((4,4),(4,3),3) (4,3),2)	
((5,3),(4,3),5 ((4,2),(4,3),5}	
⋮	((4,4),(4,3))
((4,2),(4,5),5),	((5,4),(4,4))

c) part 3 Heuristic DFS

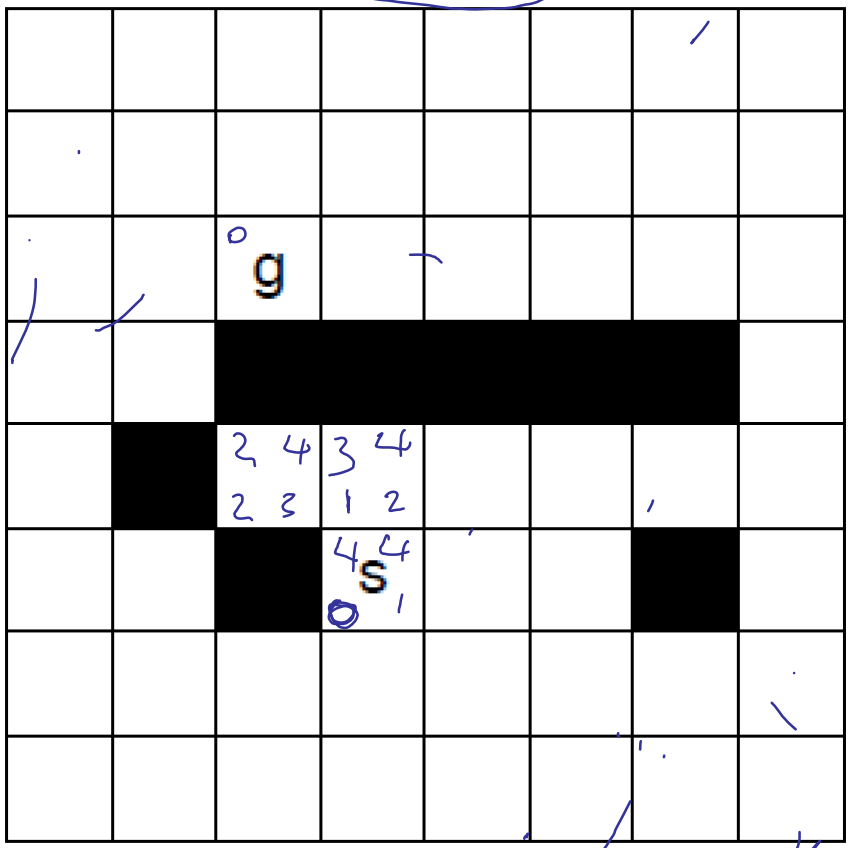
(<6,4>, <5,4>, 5),
(<5,3>, <6,4>, 5) }



(d) A* Note: Manhattan distance is admissible and monotone

Use two grids also. In the first grid, write in the f values. (A* computes these as it goes along, because the g part - distance from the start - gets updated)

For your work during your turn in one grid, with order of node expansion & path found

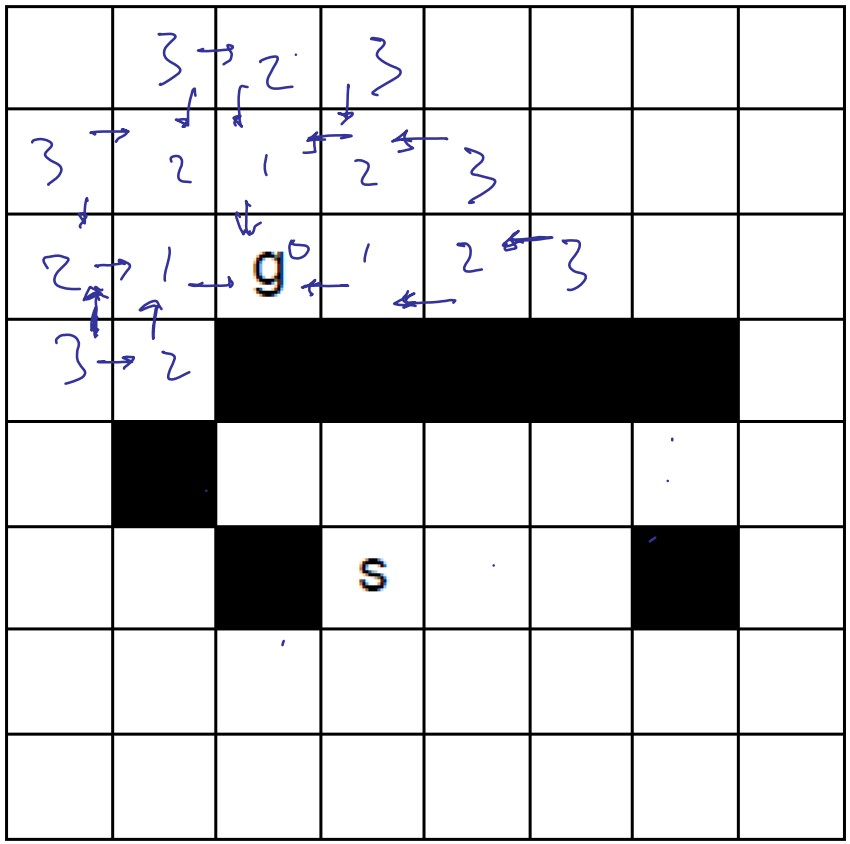


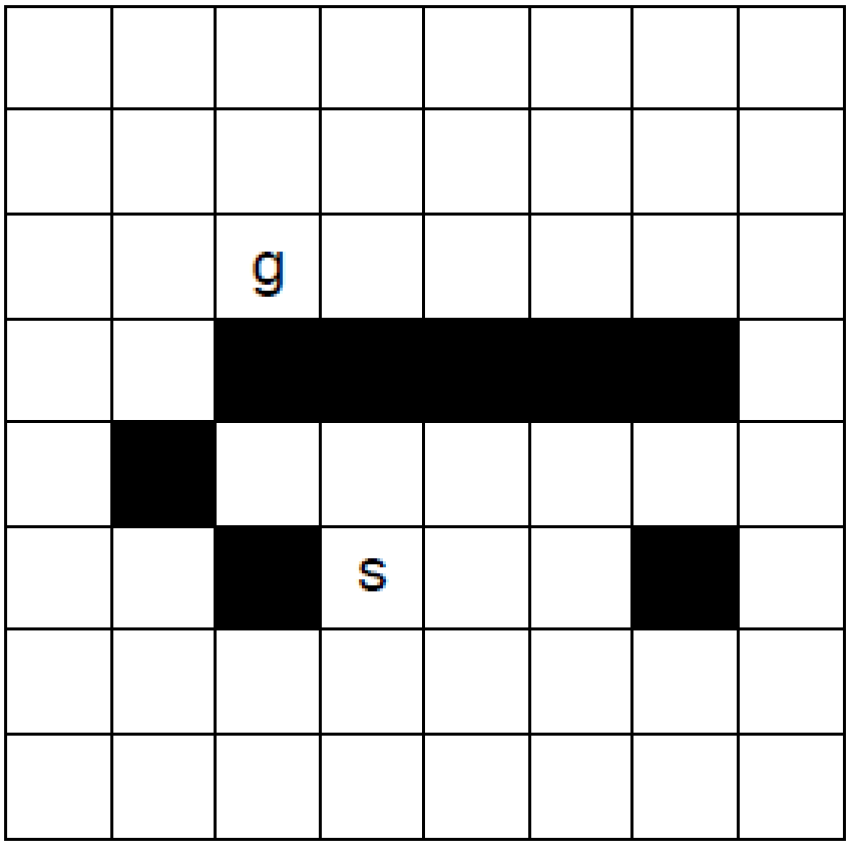
h	f
g	# of node expansion

Do this!!

(e) Dynamic programming

$$\text{cost-to-goal}(u) = \begin{cases} 0 & \text{for goal} \\ \min \text{cost}(u, w) & \end{cases}$$




g
s