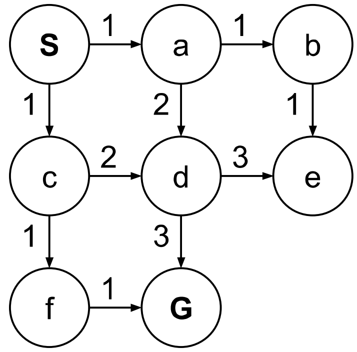
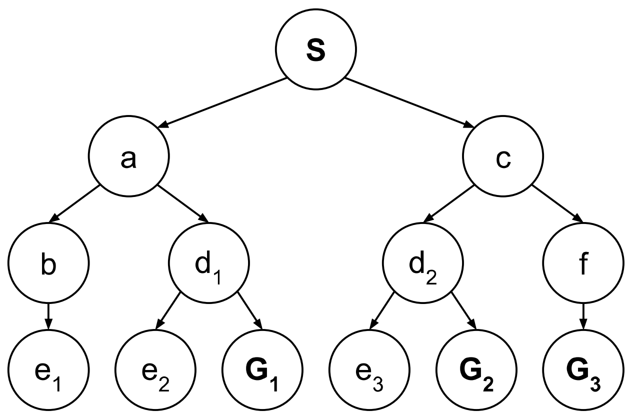
Quiz 1 - Search

Consider the following search space graph, where **S** is the starting state, and **G** the only goal state:



1. Draw the complete search tree for this problem.



2. Simulate DFS, BFS, UCS, and A* on this graph using TREE search (not graph search) and list which nodes are expanded in what order, what solution is returned for each, and whether or not this is an optimum solution. For A*, use manhattan distance from **G** as the heuristic *h* (for example *h*(*b*) = 3). For all algorithms, if there are ties for priority when popping from the fringe, break the ties using alphabetical order (earlier in the alphabet is popped first).

	Nodes expanded	Solution	Optimal?
DFS	**S** a b e_1 d_1 e_2 ($\mathbf{G_1}$)	**S** a d **G**	No
BFS	**S** a c b d_1 d_2 f e_1 e_2 ($\mathbf{G_1}$)	**S** a d **G**	No
UCS	**S** a c b f d_1 d_2 e_1 ($\mathbf{G_3}$)	**S** c f **G**	Yes
A*	**S** a c f ($\mathbf{G_3}$)	**S** c f **G**	Yes

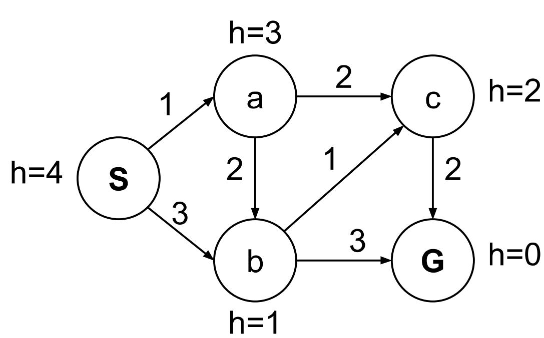
3. Simulate DFS, BFS, UCS, and A* on this graph using GRAPH search and list which nodes are expanded in what order, what solution is returned for each, and whether or not this is an optimum solution. For A*, use manhattan distance from **G** as the heuristic *h* (for example *h*(*b*) = 3). For all algorithms, if there are ties for priority when popping from the fringe, break the ties using alphabetical order (earlier in the alphabet is popped first).

	Nodes expanded	Solution	Optimal?
DFS	**S** a b e_1 d_1 (**G_1**)	**S** a d **G**	No
BFS	**S** a c b d_1 f e_1 (**G_1**)	**S** a d **G**	No
UCS	**S** a c b f d_1 e_1 (**G_3**)	**S** c f **G**	Yes
A*	**S** a c f (**G_3**)	**S** c f **G**	Yes

4. What conditions must a search problem satisfy such that BFS is guaranteed to find an optimal solution?

If the costs are all uniform and non-negative, then BFS is guaranteed to find an optimal solution.

5. For the following search problem, is the given heuristic admissible? Is it consistent? Does A* graph search return an optimal solution? (**S** is the start state, and **G** is the only goal state.)



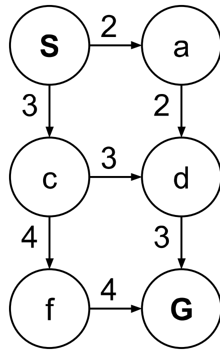
The heuristic is an underestimate of the true cost at every state, so it IS admissible.
The heuristic does not drop by more than the arc cost at each arc, so it IS consistent.
A* graph search must return an optimal solution, since the heuristic is consistent.

6. Simulate greedy search on the graph in problem 5 and list which nodes are expanded, and what solution is returned. What is the cost of this solution? Is it optimal?

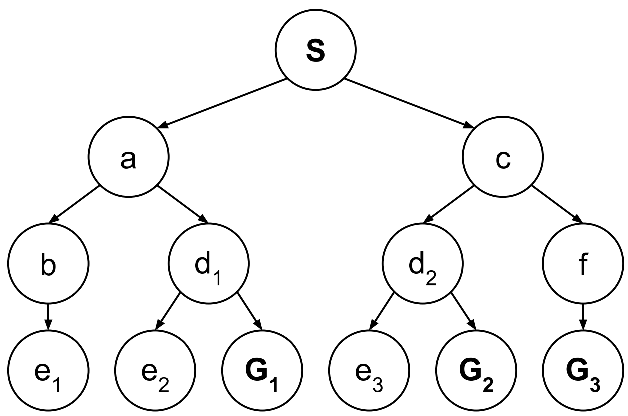
Greedy search expands the node that minimizes the heuristic.
Nodes expanded: S b (G)
Solution: S b G
Cost: 6 (Not optimal, because S a c G costs 5)

Quiz 1 - Search

Consider the following search space graph, where **S** is the starting state, and **G** the only goal state:



1. Draw the complete search tree for this problem.



2. Simulate DFS, BFS, UCS, and A* on this graph using TREE search (not graph search) and list which nodes are expanded in what order, what solution is returned for each, and whether or not this is an optimum solution. For A*, use manhattan distance from **G** as the heuristic *h* (for example *h*(*b*) = 3). For all algorithms, if there are ties for priority when popping from the fringe, break the ties using alphabetical order (earlier in the alphabet is popped first).

	Nodes expanded	Solution	Optimal?
DFS	**S** a b e_1 d_1 e_2 (**G_1**)	**S** a d **G**	Yes
BFS	**S** a c b d_1 d_2 f e_1 e_2 (**G_1**)	**S** a d **G**	Yes
UCS	**S** a b c d_1 e_1 d_2 e_2 f (**G_1**)	**S** a d **G**	Yes
A*	**S** a c d_1 b e_1 d_2 (**G_1**)	**S** a d **G**	Yes

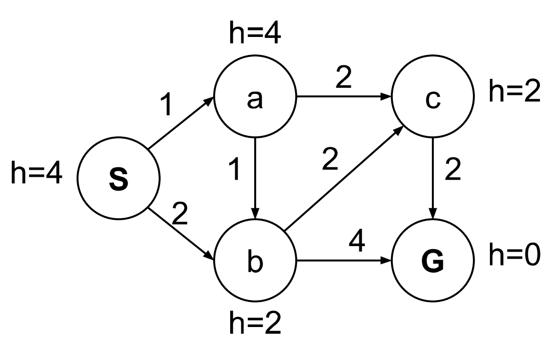
3. Simulate DFS, BFS, UCS, and A* on this graph using GRAPH search and list which nodes are expanded in what order, what solution is returned for each, and whether or not this is an optimum solution. For A*, use manhattan distance from **G** as the heuristic *h* (for example *h*(*b*) = 3). For all algorithms, if there are ties for priority when popping from the fringe, break the ties using alphabetical order (earlier in the alphabet is popped first).

	Nodes expanded	Solution	Optimal?
DFS	**S** a b e_1 d_1 (**G_1**)	**S** a d **G**	Yes
BFS	**S** a c b d_1 f e_1 (**G_1**)	**S** a d **G**	Yes
UCS	**S** a b c d_1 e_1 f (**G_1**)	**S** a d **G**	Yes
A*	**S** a c d_1 b e_1 (**G_1**)	**S** a d **G**	Yes

4. What conditions must a search problem satisfy such that BFS is guaranteed to find an optimal solution?

If the costs are all uniform and non-negative, then BFS is guaranteed to find an optimal solution.

5. For the following search problem, is the given heuristic admissible? Is it consistent? Does A* graph search return an optimal solution? (**S** is the start state, and **G** is the only goal state.)



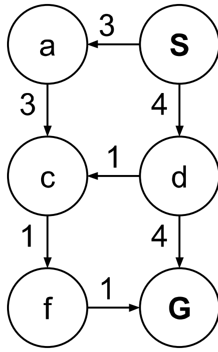
The heuristic is an underestimate of the true cost at every state, so it IS admissible.
The heuristic overestimates arc ab: h(a) - h(b) = 4 - 2 = 2 > c(ab), so it is NOT consistent.
A* graph search does return the optimal solution, despite the inconsistency.

6. Simulate greedy search on the graph in problem 5 and list which nodes are expanded, and what solution is returned. What is the cost of this solution? Is it optimal?

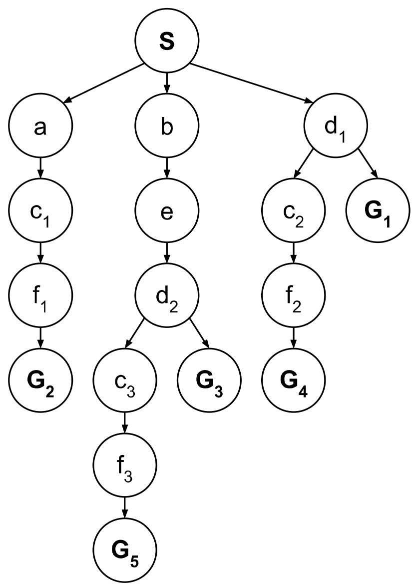
Greedy search expands the node that minimizes the heuristic.
Nodes expanded: S b (G)
Solution: S b G
Cost: 6 (Not optimal, because S a c G costs 5)

Makeup Quiz 1 - Search

Consider the following search space graph, where **S** is the starting state, and **G** the only goal state:



1. Draw the complete search tree for this problem.



2. Simulate DFS, BFS, UCS, and A* on this graph using TREE search (not graph search) and list which nodes are expanded in what order, what solution is returned for each, and whether or not this is an optimum solution. For A*, use manhattan distance from **G** as the heuristic *h* (for example *h*(*b*) = 3). For all algorithms, if there are ties for priority when popping from the fringe, break the ties using alphabetical order (earlier in the alphabet is popped first).

	Nodes expanded	Solution	Optimal?
DFS	**S** a c_1 f_1 (**G_2**)	**S** a c f **G**	No
BFS	**S** a b d_1 c_1 e c_2 (**G_1**)	**S** d **G**	No
UCS	**S** b e a d_2 c_3 d_1 c_2 f_3 c_1 f_2 (**G_5**)	**S** b e d c f **G**	Yes
A*	**S** b e d_2 d_1 a c_3 f_3 (**G_5**)	**S** b e d c f **G**	Yes

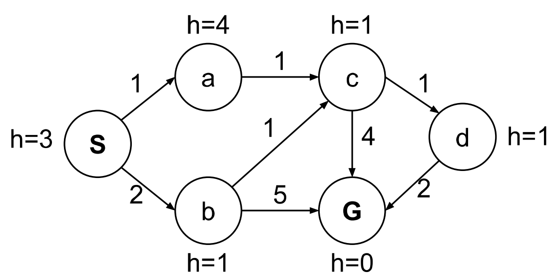
3. Simulate DFS, BFS, UCS, and A* on this graph using GRAPH search and list which nodes are expanded in what order, what solution is returned for each, and whether or not this is an optimum solution. For A*, use manhattan distance from **G** as the heuristic *h* (for example *h*(*b*) = 3). For all algorithms, if there are ties for priority when popping from the fringe, break the ties using alphabetical order (earlier in the alphabet is popped first).

	Nodes expanded	Solution	Optimal?
DFS	**S** a c_1 f_1 ($\mathbf{G_2}$)	**S** a c f **G**	No
BFS	**S** a b d_1 c_1 e ($\mathbf{G_1}$)	**S** d **G**	No
UCS	**S** b e a d_2 c_3 f_3 ($\mathbf{G_5}$)	**S** b e d c f **G**	Yes
A*	**S** b e d_2 a c_3 f_3 ($\mathbf{G_5}$)	**S** b e d c f **G**	Yes

4. What heuristic makes A* search equivalent to UCS?

The heuristic that assigns 0 to every state makes A* search equivalent to UCS.

5. For the following search problem, is the given heuristic admissible? Is it consistent? Does A* graph search return an optimal solution? (**S** is the start state, and **G** is the only goal state.)



The heuristic is an underestimate of the true cost at every state, so it IS admissible.
The heuristic overestimates arc ac: h(a) - h(c) = 4 - 1 = 3 > c(ac), so it is NOT consistent.
A* graph search returns S b c d G, which costs 6, and is NOT optimal since S a c d G is a solution of cost 5.

6. Simulate greedy search on the graph in problem 5 and list which nodes are expanded, and what solution is returned. What is the cost of this solution? Is it optimal?

Greedy search expands the node that minimizes the heuristic.
Nodes expanded: S b (G)
Solution: S b G
Cost: 7 (Not optimal, because S a c d G costs 5)