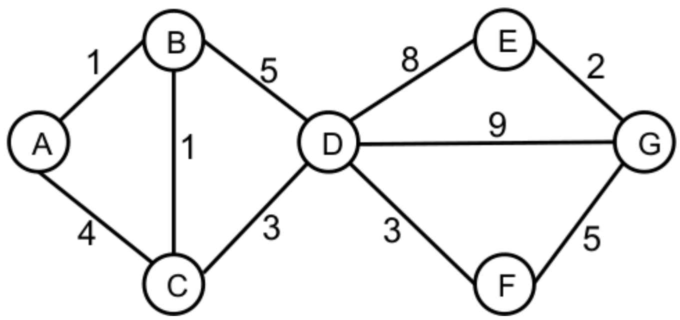
Please use LaTeX to produce your writeups. See the Homework Assignments page on the class website for details.

1 Uninformed Search

Consider the state space graph shown below. A is the start state and G is the goal state. The costs for each edge are shown on the graph. Each edge can be traversed in both directions.



Use the Graph Search Algorithm (v2) discussed in class. Execute the following search algorithms using priority queues, by filling in the search table for each part. Write nodes as a tuple containing a state sequence and cost (e.g. (A-B-C, 2)). Note that for Breadth first and Depth first the algorithms ignore the true "cost" so you can just use the depth of the node as the second part of the tuple and then expand nodes with either the highest or lowest depth. Break ties alphabetically. Note that all steps in the table below will necessarily be used. Skip any steps where a node is removed from the frontier but not expanded. Note that nodes are only expanded after they are removed from the frontier, after checking the goal test, and after checking if not in the closed set.

1. **Breadth First Graph Search.**

Step	**Priority Queue**	**Expand**
1	((A), 0)	A
2	((A-B), 1), ((A-C), 1)	B
3	((A-C), 1), ((A-B-C), 2), ((A-B-D), 2)	C
4	((A-B-D), 2)	D
5	((A-B-D-E), 3), ((A-B-D-F), 3), ((A-B-D-G), 3)	E
6	((A-B-D-F), 3), ((A-B-D-G), 3), ((A-B-D-E-G), 4)	F
7	((A-B-D-G), 3), ((A-B-D-E-G), 4), ((A-B-D-F-G), 4)	
8		

Solution: A-B-D-G

2. **Depth First Graph Search.**

Step	Priority Queue	Expand
1	((A), 0)	A
2	((A-B), 1), ((A-C), 1)	B
3	((A-B-C), 2), ((A-B-D), 2), ((A-C), 1)	C
4	((A-B-C-D), 3),	D
5	((A-B-C-D-E), 4), ((A-B-C-D-F), 4), ((A-B-C-D-G), 4), ((A-B-D), 2), ((A-C), 1)	E
6	((A-B-C-D-E-G), 5),((A-B-C-D-F), 4), ((A-B-C-D-G), 4), ((A-B-D), 2), ((A-C), 1)	
7		
8		

Solution: A-B-C-D-E-G

3. **Uniform Cost Graph Search.**

Step	Priority Queue	Expand
1	((A), 0)	A
2	((A-B), 1), ((A-C), 4)	B
3	((A-B-C), 2), ((A-C), 4), ((A-B-D), 6)	C
4	((A-B-C-D), 5), ((A-B-D), 6)	D
5	((A-B-C-D-F), 8), ((A-B-C-D-E), 13), ((A-B-C-D-G), 14)	F
6	((A-B-C-D-E), 13), ((A-B-C-D-F-G), 13), ((A-B-C-D-G), 14)	E
7	((A-B-C-D-F-G), 13), ((A-B-C-D-G), 14), ((A-B-C-D-E-G), 15)	
8		

Solution: A-B-C-D-F-G

2 Heuristic Search

1. Consider the two heurisitics h_1 and h_2, only one of which is consistent. Which one is consistent?

Node	A	B	C	D	E	F	G
h_1	9.5	9	8	7	1.5	4	0
h_2	10	12	10	8	1	4.5	0

h_1 is consistent

2. Then do A* search with that heuristic.

Step	Priority Queue	Expand
1	((A), 0 + 9.5)	A
2	((A-B), 1 + 9), ((A-C), 4 + 8)	B
3	((A-B-C), 2 + 8), ((A-B-D), 6 + 7), ((A-C), 4 + 8)	C
4	((A-B-C-D), 5 + 7), ((A-B-D), 6 + 7)	D
5	((A-B-C-D-E), 13 + 1.5), ((A-B-C-D-F), 8 + 4), ((A-B-C-D-G), 14 + 0), ((A-B-D), 6 + 7)	F
6	((A-B-C-D-E), 13 + 1.5), ((A-B-C-D-F-G), 13 + 0), ((A-B-C-D-G), 14 + 0)	
7		
8		

Solution: A-B-C-D-F-G

3. Suppose you are completing the new heuristic function h_3 shown below. All the values are fixed except $h_3(B)$.

Node	A	B	C	D	E	F	G
h_3	10	?	9	7	1.5	4.5	0

For each of the following conditions, write the set of values that are possible for $h_3(B)$. For example, to denote all non-negative numbers, write [0, ∞], to denote the empty set, write $\emptyset$, and so on.

(a) What values of $h_3(B)$ make h_3 admissible?

For any heuristic to be admissible, it must not overestimate the cost to reach the goal. Since the cost to get from B to the end is 12, any number in [-∞, 12] satisfies the constraint.

(b) What values of $h_3(B)$ make h_3 consistent?

For this question, we need to satisfy the following $10 \leq 1 + h_3(B)$, $h_3(B) \leq 1 + 9$, and $h_3(B) \leq 5 + 7$. Simplifying these, we find $9 \leq h_3(B) \leq 10$ so x is in [9, 10].

(c) What values of $h_3(B)$ will cause A* graph search to expand from node A to C, then node A to B, then node A to B to D in that order?

In order for A* graph search to expand in the order specified above, $h_3(B)$ would have to satisfy $1 + h_3(B) > 13$ for it to expand A-C first. To then expand A-B, $h_3(B)$ would have to satisfy $1 + h_3(B) < 14$. Simplifying here, $12 < x < 13$ so x is in (12, 13). Depending on how we break ties, we might be able to give hard bounds on either side.