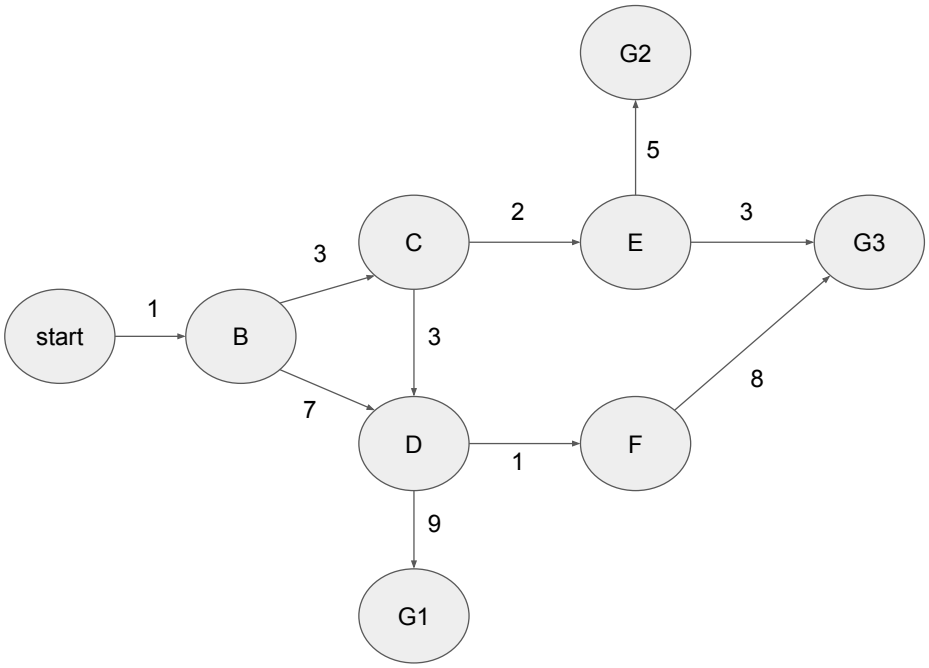
1 Uninformed Search

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. Edges are directed and can only be traversed in the direction indicated by the arrow. When doing Breadth and depthfirst search you can just use the depth of the node as the cost.



Execute the following search algorithms using priority queues, by filling in the search table for each part. Break ties alphabetically. Note that all steps in the table below will necessarily be used. You may skip any steps where a node is removed from the fringe but not expanded. Note that nodes are only expanded after they are removed from the fringe, after checking the goal test, and after checking if not in the closed set.

1. **Breadth First Graph Search.**

Step	Priority Queue (a.k.a. the Fringe)	Expand
1	(S,0)	S
2	(S-B,1)	B
3	(S-B-C,2), (S-B-D,2)	C
4	(S-B-D,2), (S-B-C-D,3), (S-B-C-E,3)	D
5	(S-B-C-E,3), (S-B-D-G1,3), (S-B-D-F,3)	E
6	(S-B-C-E-G2,4), (S-B-C-E-G3,4), (S-B-D-G1,3), (S-B-D-F,3)	F
7	(S-B-C-E-G2,4), (S-B-C-E-G3,4), (S-B-D-G1,3), (S-B-D-F-G3,4)	

Solution: S-B-D-G1

Note that in step 5, element (S-B-C-D,3) is not expanded since D is in the closed list. It is removed from the Fringe (as noted above you can skip steps that don't result in a node being expanded) and then E is expanded.

Note also that in step 7, nothing is expanded since the goal node is removed from the fringe and the goal check happens before expansion.

2. **Depth First Graph Search.**

Step	**Priority Queue**	**Expand**
1	(S,0)	S
2	(S-B,1)	B
3	(S-B-C,2), (S-B-D,2)	C
4	(S-B-C-E,3), (S-B-C-D,3), (S-B-D,2)	D
5	(S-B-C-E,3), (S-B-C-D-G1,4), (S-B-C-D-F,4), (S-B-D,2)	F
6	(S-B-C-E,3), (S-B-C-D-G1,4), (S-B-C-D-F-G3,5), (S-B-D,2)	
7		
8		

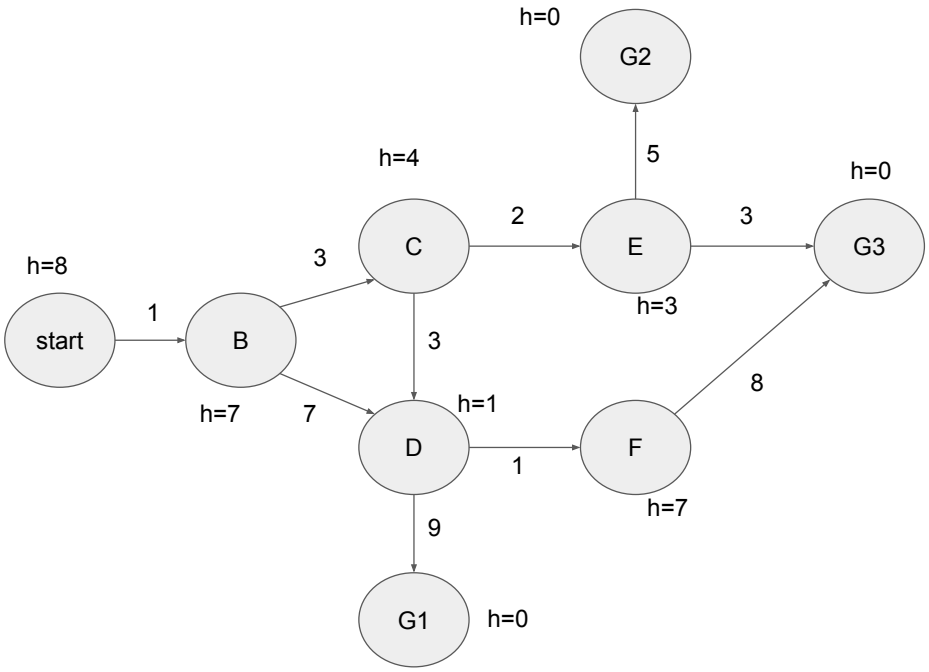
Solution: S-B-C-D-F-G3

3. **Uniform Cost Graph Search.**

Step	**Priority Queue**	**Expand**
1	(S,0)	S
2	(S-B,1)	B
3	(S-B-C,4), (S-B-D, 8)	C
4	(S-B-C-E,6), (S-B-C-D,7), (S-B-D, 8)	E
5	(S-B-C-E-G2, 11), (S-B-C-E-G3, 9) (S-B-C-D,7), (S-B-D, 8)	D
6	(S-B-C-E-G2, 11), (S-B-C-E-G3, 9) (S-B-C-D-G1,16), (S-B-C-D-F,8)	F
7	(S-B-C-E-G2, 11), (S-B-C-E-G3, 9) (S-B-C-D-G1,16), (S-B-C-D-F-G3,16)	
8		

Solution: S-B-C-E-G3

Consider the following heuristic function h:



1. Is the heuristic admissible? Yes.
2. Is the heuristic consistent? Yes.
3. Solve using best-first search (greedy)

Step	**Priority Queue**	**Expand**
1	(S,8)	S
2	(S-B,7)	B
3	(S-B-C, 4), (S-B-D, 1)	D
4	(S-B-C, 4), (S-B-D-F, 7), (S-B-D-G1, 0)	

Solution: S-B-D-G1

4. Solve using A-star search

Step	Priority Queue	Expand
1	(S, 0 + 8)	S
2	(S-B, 1 + 7)	B
3	(S-B-C, 4 + 4), (S-B-D, 8 + 1)	C
4	(S-B-C-D, 7 + 1), (S-B-C-E, 6 + 3) (S-B-D, 8+1)	D
5	(S-B-C-D-F, 8 + 7), (S-B-C-D-G1, 16 + 0), (S-B-C-E, 6 + 3)	E
6	(S-B-C-D-F, 8 + 7), (S-B-C-D-G1, 16 + 0), (S-B-C-E-G2, 11 + 0), (S-B-C-E-G3, 9 + 0)	

Solution: S-B-C-E-G3

Note that after step 4 we try to expand D again (by removing from the fringe the node (S-B-D, 8+1)), but find it in the closed set so we remove (S-B-D, 8+1) from the fringe and then pop and expand E in step 5.

Note that we get the same solution as uniform cost search, but we expanded one fewer node. In general, for the types of large statespaces we typically apply A-star, we will see a much larger decrease in the number of node expansions when using A-star.