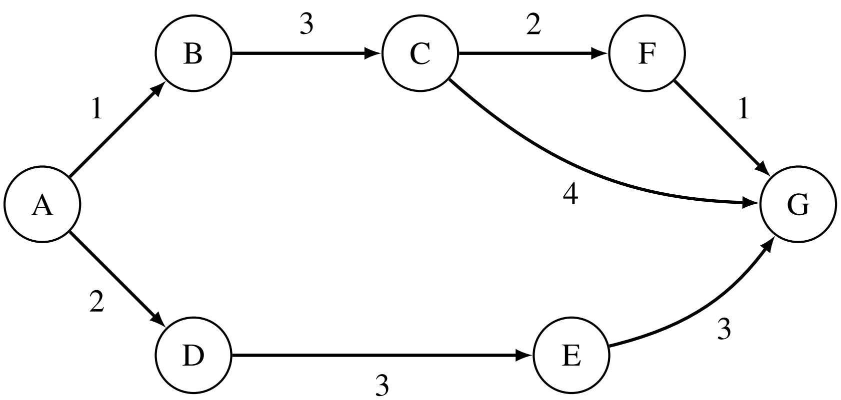
1 A* Search

For the graph below, let `g(u, v)` be the weight of the edge between any nodes `u` and `v`. Let `h(u, v)` be the value returned by the heuristic for any nodes `u` and `v`.

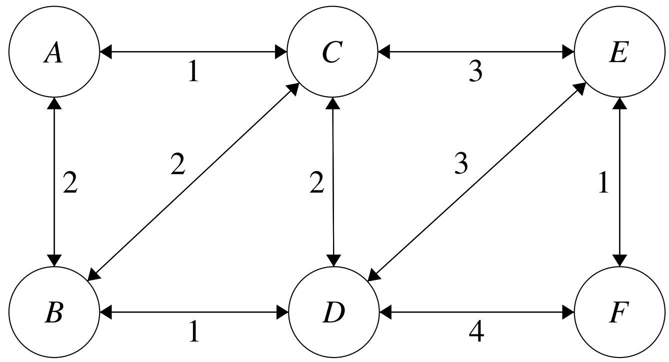



```
Edge weights:      Heuristics:
g(A, B) = 1        h(A, G) = 8
g(B, C) = 3        h(B, G) = 6
g(C, F) = 2        h(C, G) = 5
g(C, G) = 4        h(F, G) = 1
g(F, G) = 1        h(D, G) = 6
g(A, D) = 2        h(E, G) = 3
g(D, E) = 3
g(E, G) = 3
```

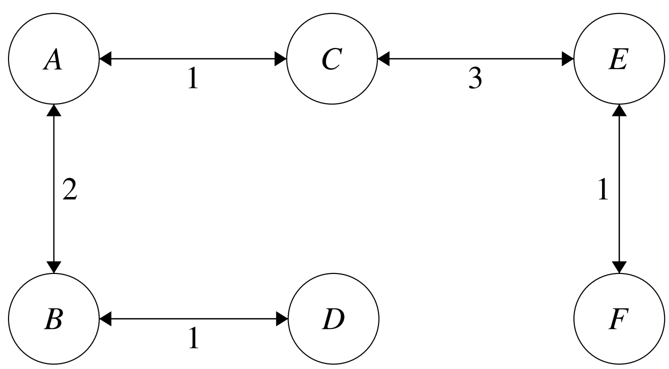
a) Given the weights and heuristic values for the graph below, what path would A* search return, starting from A and with G as a goal? A* would return A-D-E-G.

b) Is the heuristic admissible? Why or why not? The heuristic is not admissible because h(C, G) = 5, but the shortest path from C to G has length 3.

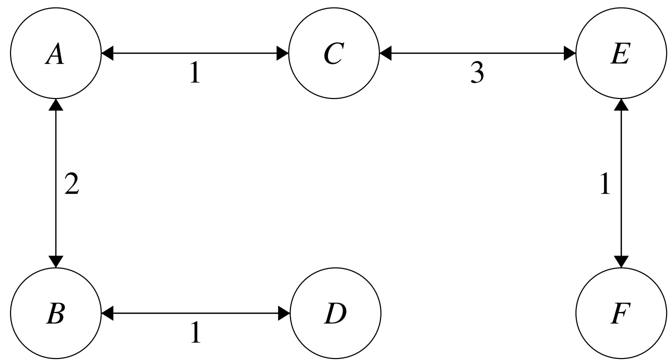
2 Minimum Spanning Trees



a) Perform Prim's algorithm to find the minimum spanning tree of the above graph. Pick A as the initial node. Whenever there are more than one node with the same cost, process them in alphabetical order.



b) Use Kruskal's algorithm to find a minimum spanning tree. Is it the same as the one found by Prim's?



It happens to be the same here, but it is also possible for Kruskal's to return a different MST based on how its sorting breaks ties.

c) Draw the resulting WQU tree at the end of the execution of Kruskal's. When there are ties, choose the root to be the alphabetically first node. (There may be multiple possible answers)