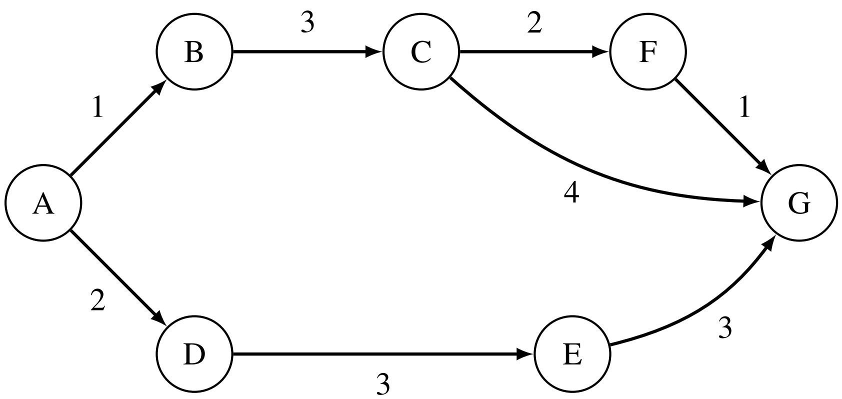
CS 61B A* Search and MSTs Spring 2020

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`. Remember the heuristic serves to estimate the distance between two nodes `u` and `v`.

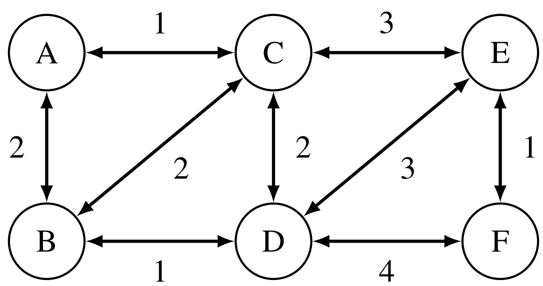



```
Edge weights:    Heuristics:
g(A, B) = 1      h(A, G) = 8
g(B, C) = 3      h(B, G) = 6
g(C, F) = 2      h(C, G) = 5
g(C, G) = 4      h(F, G) = 1
g(F, G) = 1      h(D, G) = 6
g(A, D) = 2      h(E, G) = 3
g(D, E) = 3
g(E, G) = 3
```

(a) Given the weights and heuristic values for the graph above, what would A* search return as the shortest path from A to G?

(b) Is the heuristic admissible? Why or why not? A heuristic is admissible if it never overestimates the distance it is estimating.

2 Minimum Spanning Trees



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

(b) Use Kruskal's algorithm to find a valid minimum spanning tree of the graph above. If there are multiple edges with the same cost, process them in alphabetical order. Will Prim's and Kruskal's always return the same MST?

(c) Draw the final state of the tree that results from union find operations executed when running Kruskal's on the graph above. When selecting a new root, break ties alphabetically.