1 Dijkstra's Algorithm

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`.

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

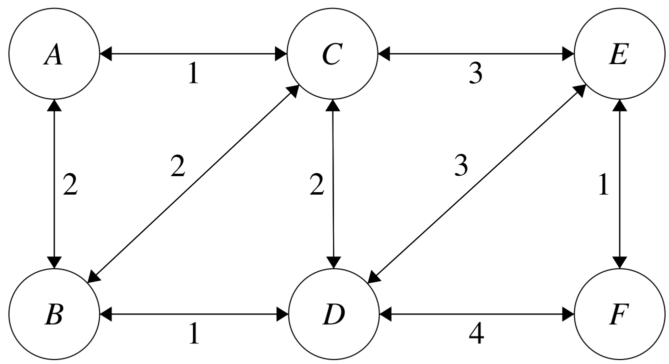
a) Run Dijkstra's Algorithm to find the shortest paths from A to every other vertex. You may find it helpful to keep track of the priority queue and make a table of current distances.

2 Extra for Experts

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?

b) Is the heuristic admissible? Why or why not?

3 Minimum Spanning Trees



a) Perform Prim's algorithm to find the minimum spanning tree of the following 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.

c) There are quite a few MSTs here. How many can you find?

4 Sorting I

Show the steps taken by each sort on the following unordered list:

`106, 351, 214, 873, 615, 172, 333, 564`

(a) Insertion sort. Show the sorted and unsorted portions at every step.

(b) Selection sort. Show the sorted and unsorted portions at every step.

(c) Merge sort. Show the sorted and unsorted portions at every step.

(c) Use heapsort to sort the following array (hint: draw out the heap). Draw out the array at each step:

`106, 615, 214, 873, 351`

5 Sorting II

Match the sorting algorithms to the sequences, each of which represents several intermediate steps in the sorting of an array of integers.
Algorithms: Heapsort, insertion sort, selection sort, mergesort

(a) `12, 32, 14, 34, 17, 38, 23, 11`
`32, 17, 23, 11, 12, 14, 34, 38`

(b) `23, 45, 12, 4, 65, 34, 20, 43`
`12, 23, 45, 4, 65, 34, 20, 43`

(c) `12, 7, 8, 4, 10, 2, 5, 34, 14`
`2, 4, 5, 7, 8, 12, 10, 34, 14`