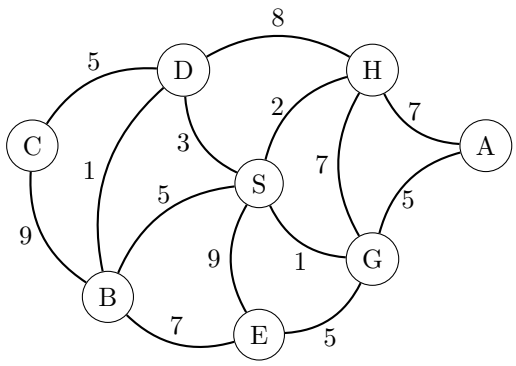
Week 11. Problem set

1. Run Prim-Jarník algorithm [CLRS, §21.2] on the following graph $\mathcal{G}$, starting at vertex S, and answer the questions below.



   (a) Draw a minimum spanning tree of the graph $\mathcal{G}$.

   (b) Is the minimum spaning tree for graph $\mathcal{G}$ unique?

   (c) Assuming that the algorithm is using Binary Heap implementation [CLRS, §6] of a priority queue, show the state of the priority queue after each iteration of the algorithm (i.e. after adding each new vertex to the MST). The graph contains 8 vertices, which means that your solution must provide 8 states of the Binary heap. Each heap state must be represented as an array.

   For example, a binary min-heap containing key-value pairs $\langle 3, a\rangle, \langle 2, b\rangle, \langle 1, c\rangle, \langle 4, d\rangle$ may be represented as follows:

$1, c$	$3, a$	$2, b$	$4, d$

2. Suppose that T is the minimum spaning tree for graph $\mathcal{G}$. How difficult is it to update the minimum spanning tree after adding a new vertex and its incident edges to $\mathcal{G}$.

   (a) Describe your algorithm (write pseudocode).

   (b) Provide the asymptotic time complexity for your algorithm.

   (c) Justify the asymptotic time complexity (1–3 sentences).

3. Suppose that all edge weights in a graph are *real numbers* uniformly distributed in the range from 0 to k.

   (a) Which of the MST algorithms, Kruskal's or Prim's (possibly with minor modifications) can run faster on such a graph?

   (b) Which modification is required to make the algorithm run faster? Justify your answer (2–3 sentences).

   (c) What is the *average* case time complexity of the modified algorithm? Justify your answer (1—2 sentences).

References

[CLRS] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2022. *Introduction to algorithms, Fourth Edition*. MIT press.

[GTG] M. T. Goodrich, R. Tamassia, and M. H. Goldwasser. *Data Structures and Algorithms in Java*. WILEY 2014.