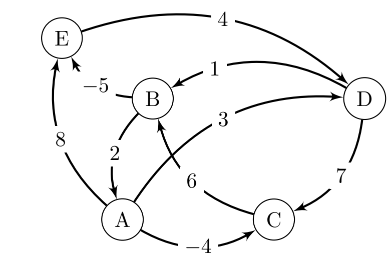
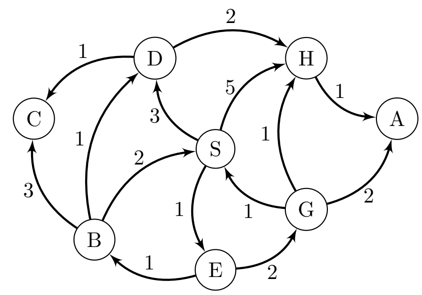
Week 12. Problem set

1. Run the Floyd-Warshall algorithm [CLRS, §23.2] on the following graph.

   Consider vertices alphabetically (A, B, C, D, E). Show the state of distance matrix D after each iteration of outer loop in the algorithm. Since the graph has 5 vertices, you must provide five 5×5 matrices in your answer. No justification is required.



2. Run Dijkstra's algorithm [CLRS, §22.3] on the following graph $\mathcal{G}$, starting at vertex S, and answer the questions below.



   (a) Draw a shortest paths tree of the graph $\mathcal{G}$.

   (b) Is the shortest paths 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 shortest paths tree). 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:

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

3. Consider a graph $\mathcal{G} = (V, E)$ that represents a computer network. Each edge (u, v) represents a physical cable connecting two computers and has a *length* $\ell(u, v)$ and a bandwidth $b(u, v)$.

   The *length of a path* P is computed as the sum of lengths of all edges in P. The *bandwidth of a path* P is computed as the minimum of bandwidth values for all edges in P.

   Given two paths P_1, P_2 in the $\mathcal{G}$ with lengths ℓ_1, ℓ_2 and bandwidth values b_1, b_2, we consider P_1 to be *better* than P_2 when $\ell_1 < \ell_2$ or ($\ell_1 = \ell_2$ and $b_1 > b_2$).

   Consider a modification of Dijkstra's algorithm [CLRS, §22.3] that finds the best paths in such a computer network from a single source vertex. Answer the following questions:

   (a) What should serve as key (priority) in the priority queue in the modified Dijkstra's algorithm?

   (b) Is it still possible to rely on DECREASE-KEY in the implementation? If yes, explain how. If no, explain why. In either case provide a brief justification (1–2 sentences).

   (c) Correctness of Dijkstra's algorithm [CLRS, Theorem 22.6] has a constraint on the weight function w. What constraint on functions ℓ and b is necessary for the correctness of the modified Dijkstra's algorithm?

References

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