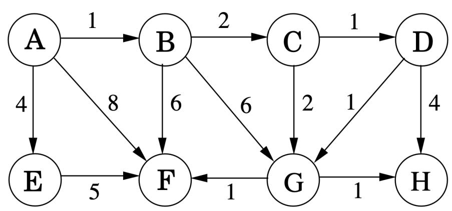
CMPSC 465	Data Structures & Algorithms	
Fall 2025	Ke Chen and Yana Safonova	Worksheet 6

Monday, Oct 13, 2025

1. Dijkstra's. Suppose Dijkstra's Algorithm is run on the following graph, starting at node A.



(i) Draw a table showing the intermediate distance values of all the nodes at each iteration of the algorithm.

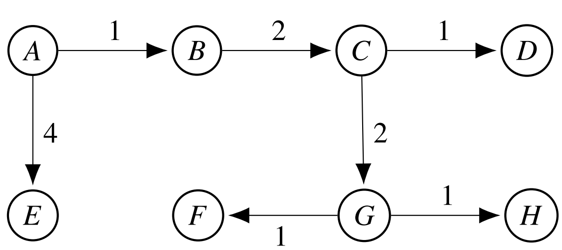
(ii) Show the final shortest-path tree.

Solution:

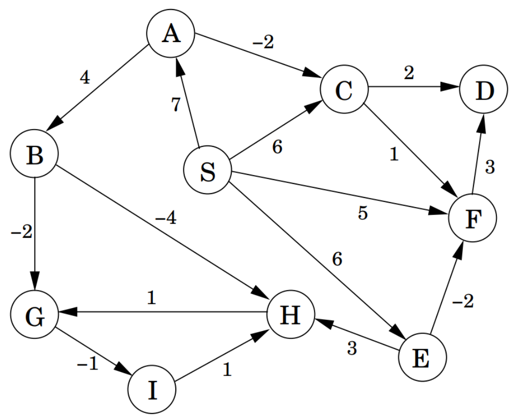
(i)

Node	Iteration 0	1	2	3	4	5	6	7
A	0	0	0	0	0	0	0	0
B	∞	1	1	1	1	1	1	1
C	∞	∞	3	3	3	3	3	3
D	∞	∞	∞	4	4	4	4	4
E	∞	4	4	4	4	4	4	4
F	∞	8	7	7	7	7	6	6
G	∞	∞	7	5	5	5	5	5
H	∞	∞	∞	∞	8	8	6	6

(ii)



2. Bellman-Ford. Suppose Bellman-Ford is used to find all the shortest paths from node S.



(i) Draw a table showing the intermediate distance values of all the nodes at each iteration of the algorithm.

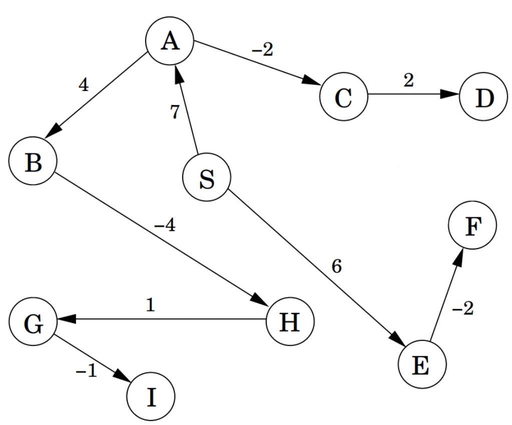
(ii) Show the final shortest-path tree.

Solution:

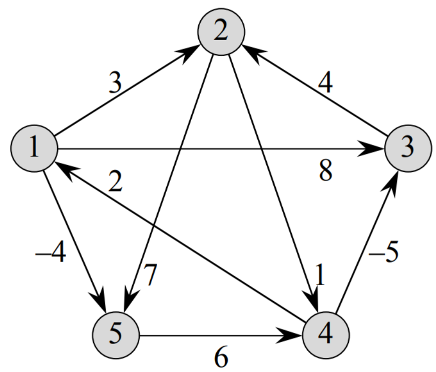
(i)

	Iteration						
Node	0	1	2	3	4	5	6
S	0	0	0	0	0	0	0
A	∞	7	7	7	7	7	7
B	∞	∞	11	11	11	11	11
C	∞	6	5	5	5	5	5
D	∞	∞	8	7	7	7	7
E	∞	6	6	6	6	6	6
F	∞	5	4	4	4	4	4
G	∞	∞	∞	9	8	8	8
H	∞	∞	9	7	7	7	7
I	∞	∞	∞	∞	8	7	7

(ii) Note that edge FD could be included instead of edge CD. Which one is chosen is dependent on the order the edges are updated.



3. Floyd-Warshall. Run Floyd-Warshall to find all pairs of shortest paths in the following graph. Show the distance matrix for each step of the algorithm, including the initial and final matrices.



Solution:

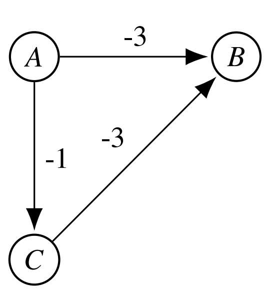
Let $_k$ be the distance matrix after step k, where the entry in row i and column j is the distance from node i to node j.

$$dist_0 = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \quad dist_1 = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \quad dist_2 = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

$$dist_3 = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \quad dist_4 = \begin{bmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix} \quad dist_5 = \begin{bmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix}$$

4. Dijkstra's with Negative Edges. Professor F. Lake suggests the following algorithm for finding the shortest path from node s to node t in a directed graph with some negative edges: add a large constant to each edge weight so that all the weights become positive, then run Dijkstra's algorithm starting at node s, and return the shortest path found to node t. Is this a valid method? Either prove that it works correctly or give a counterexample.

Solution: Counter Example:



As per the algorithm, we can add +4 to each edge. The shortest path from A to B in this new graph will be A→B with edge weight 1. However, the shortest path from A to B in the original graph is A→C→B. In general, this algorithm overly favors paths with fewer edges. A path with a single edge has its total length increased by the large constant, but a path with ten edges has its total length increased by ten times the constant.

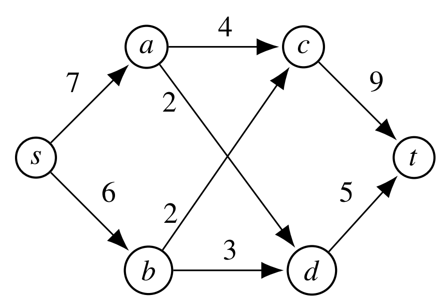
5. Network Flow. Answer the following questions on the given flow network $G = (V, E)$:

(a) Consider a function $f : E \to \mathbb{N}$ defined by $f(s,a) = 6$, $f(s,b) = 5$, $f(a,c) = 5$, $f(a,d) = 2$, $f(b,c) = 2$, $f(b,d) = 3$, $f(c,t) = 8$, and $f(d,t) = 5$. Is it a valid flow on G? If not, list all the violations and fix them.

(b) It turns out the above (after your fixes) flow f is a maximum flow in this network. We call an edge in the network a *bottleneck edge* if increasing its capacity results in an increase in the maximum flow, namely, f can improved to have a larger value. List all bottleneck edges in the network G.

(c) Give an example of a network that has no bottleneck edges.

G:



Answer:

(a) It is not a valid flow.

- $f(a,c) = 5 > 4 = c(a,c)$ violates the capacity condition; node a also violates the flow conservation because it has 6 in and $5 + 2 = 7$ out; setting $f(a,c) = 4$ fixes both.
- node c violates the flow conservation because it has $4 + 2 = 6$ in (after the above fix) and 8 out; setting $f(c,t) = 6$ fixes it.

(b) (a,c) and (b,c) are bottleneck edges.

(c) Consider the following example. Increasing the capacity of any one edge does not result in an increase of the maximum flow.