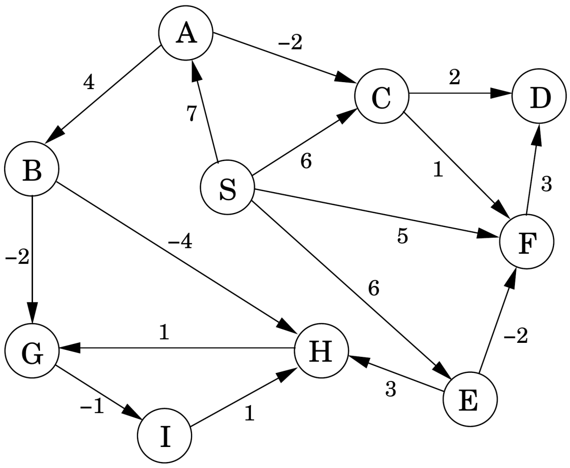
6. tutorial

1. Negative edges $+k$? Can we get rid of negative edges by adding some number $k \in \mathbb{N}$ to all edge lengths, so that we can use Dijkstra's algorithm to find the shortest path even in a graph with negative edges?

2. Likeliest path. Imagine a computer network described by a directed graph, whose vertices correspond to routers and edges are links between them. For each link we have a probability of it being functional. The probability that a certain path is functional is the product of the probabilities of individual links along this path. How to find a path between two routers which is least likely to fail (most likely to work)?

3. Generalized Shortest Paths. You are given a directed graph $G = (V, E)$ with edge lengths $\ell : E \to \mathbb{R}$ and vertex costs $c : V \to R$. For instance, imagine that this is a computer network and there is not only a delay for passing through a network cable, but also a delay incurred by passing through a router. The length of a path $v_0, e_0, v_1, e_1, \ldots, e_k, v_{k+1}$ is $(\sum_{i=0}^{k} c(v_i)) + (\sum_{i=0}^{k+1} \ell(e_i))$. How to find the shortest paths from $s \in V$ to all other nodes in G?

4. Bellman-Ford. Suppose Bellman-Ford's algorithm is run on the following graph from node A:



Draw a table displaying the intermediate values h (the distance estimates for the nodes), and draw the final shortest-path tree.

5. All-pairs, through v_0. You are given a strongly connected directed graph $G = (V, E)$ with positive edge lengths along with a particular node $v_0 \in V$. Give an efficient algorithm for finding shortest paths between all pairs of nodes, with the one restriction that these paths must all pass through v_0.

6. Dijkstra with Small Lengths. Suppose we want to run Dijkstra's algorithm on a graph whose edge weights are integers in the range $0, 1, \ldots, W$, where W is a relatively small number.

(1) Show how Dijkstra's algorithm can be made to run in time $\mathcal{O}(W|V| + |E|)$.
(2) Show an alternative implementation that takes time just $\mathcal{O}((|V| + |E|) \log W)$.

7. Max-min tunnel. Let's have a map of a city represented by directed graph. Each edge is labeled by its clearance – what is the highest truck which can pass on this road? That is, given a path, the maximum height of a truck which can pass through the path is determined by the minimum clearance along the path. Given two vertices s, t, how to find a path which allows the tallest load to pass through (i.e., a path with the maximum minimum clearance)?

8. Shortest-paths Tree? You are given a directed graph $G = (V, E)$ with (possibly negative) weighted edges, along with a specific node $s \in V$ and a tree $T = (V, E_0)$, $E_0 \subseteq E$. Give an algorithm that checks whether T is a shortest-path tree for G with starting point s. Your algorithm should run in linear time.

9. Floyd-Warshall and Shortest Cycle. Modify Floyd-Warshall's algorithm to detect, for each vertex v, the shortest cycle in G containing v. (Assume G has no negative cycles.)

10. Edges on Shortest Paths. Construct an algorithm which detects all edges lying on at least one shortest path from s to any vertex, or from s to a particular vertex t.

11. Critical Edges. Say that an edge e is *critical for s* if removing it from G changes the distances from $s \in V$, that is, there exists a v such that $d_G(s,v) < d_{G-e}(s,v)$. Design an algorithm which detects all edges critical for s.