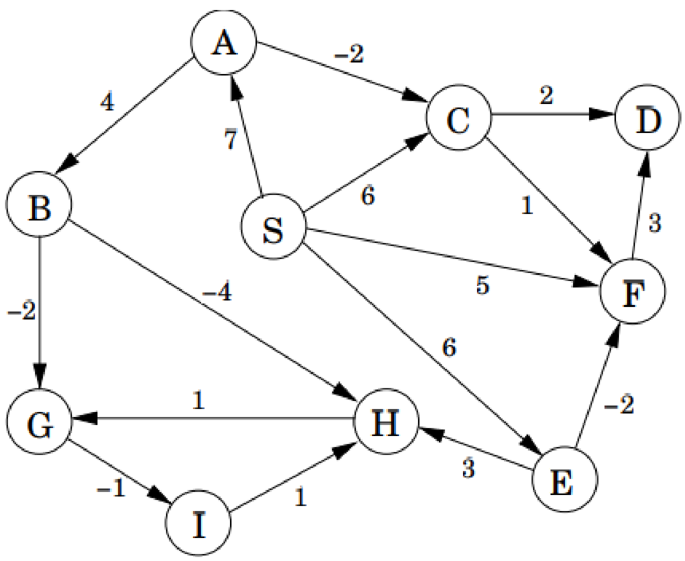
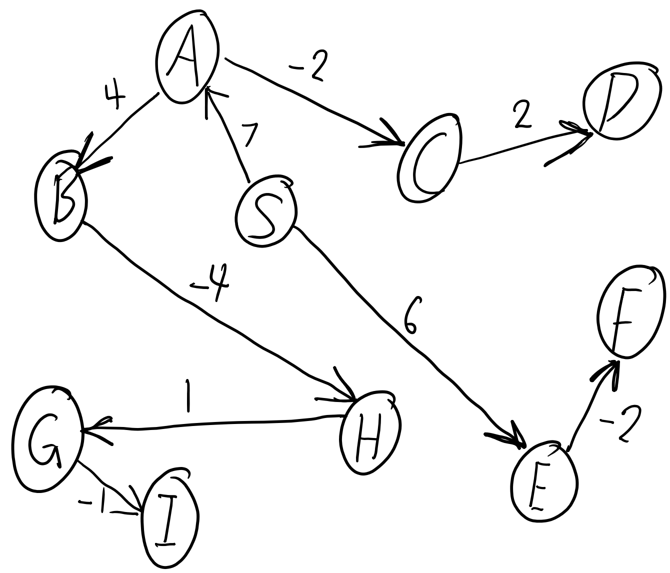
Ryne Kolessar

CSC 321 – Algorithms   <u>Assignment #4</u>

11/4/2019

1. (10 points) Textbook, page 120, exercise 4.2. You only need to show the contents of the table as in Figure 4.14 in the textbook.

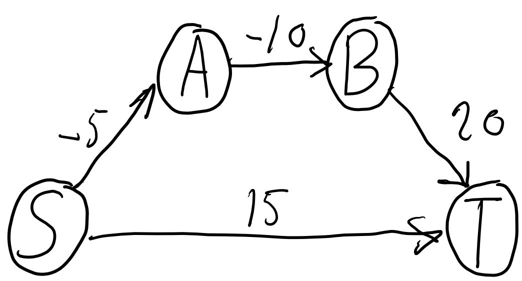




dist	S	A	B	C	D	E	F	G	H	I
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	7	∞	6	∞	6	5	∞	∞	∞
2	0	7	∞	6	8	6	5	∞	∞	∞
3	0	7	11	5	7	6	5	∞	∞	∞
4	0	7	11	5	7	6	5	9	∞	∞
5	0	7	11	5	7	6	5	9	7	∞
6	0	7	11	5	7	6	5	8	7	∞
7	0	7	11	5	7	6	5	8	7	7
8	0	7	11	5	7	6	4	8	7	7

2. (10 points) Textbook, page 121, exercise 4.8.
   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.

   This will not work, for example in this graph,



   If 10 is added to each edge to make them non-negative, then the shortest path would be S -> T

   When actually, the shortest path is S -> A -> B -> T

3. (10 points) Textbook, page 121, exercise 4.11. (Hint. Consult the running timetable on page 114 for Dijkstra's algorithm for different implementations, and the discussion below it.)
   Give an algorithm that takes as input a directed graph with positive edge lengths and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O(|V|^3)$.

   Since the graph has a cycle, you have to find two points that each have a path to each other.
   Then find the distances from each point to the other and add them together.
   Run Dijkstra from each vertex in the graph. Then check every combination of vertices in order to find the shortest cycle.