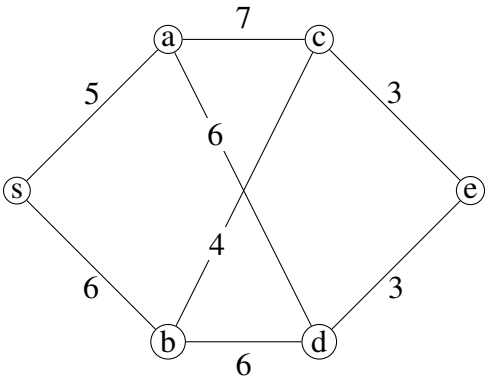
MATH F113X: Dijkstra's Algorithm

Dijkstra's Algorithm

input: a graph with distances (weights) on the edges and a starting vertex, say s
output: the shortest distance between s and every vertex in the graph
rough strategy: All vertices get **tentative** distances to vertex s. One-by-one, vertices are explored and tentative distances are updated until minimum distances are obtained.
Steps:

1. (Initialization Step) Set the tentative distance to be zero for s and ∞ for all other vertices.
2. (Iterative Step) Find the vertex, say x, with the smallest tentative distance *that has not already been explored*. (Break any ties alphabetically.)
   (a) For every edge between x and a vertex (say y) *that has not already been explored*, calculate the sum of the distance to x plus the distance along the edge to y. If this sum is **smaller** than the tentative distance at y, replace the tentative distance with the smaller value. Otherwise, leave the tentative distance of y unchanged.
   (b) Mark x as **explored** and record its tentative distance to s as its **minimum** distance to s.
   (c) If x is the last vertex then terminate the algorithm, otherwise return to the beginning of the iterative step.



Explored?	vertices	tentative distances

vertex	minimum distance to s

Think of an application of Dijkstra's Algorithm.