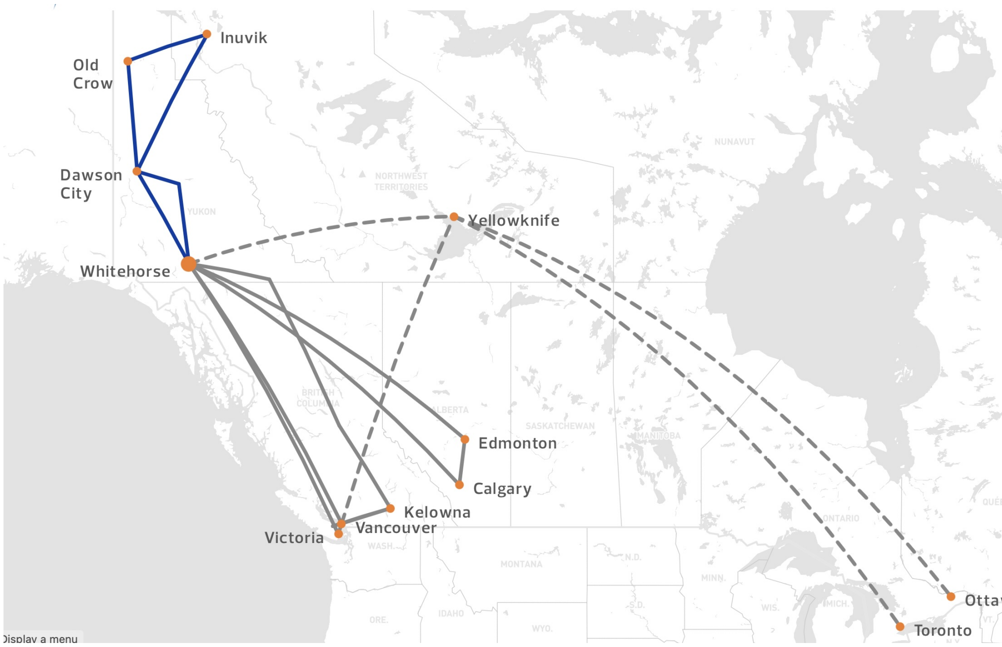
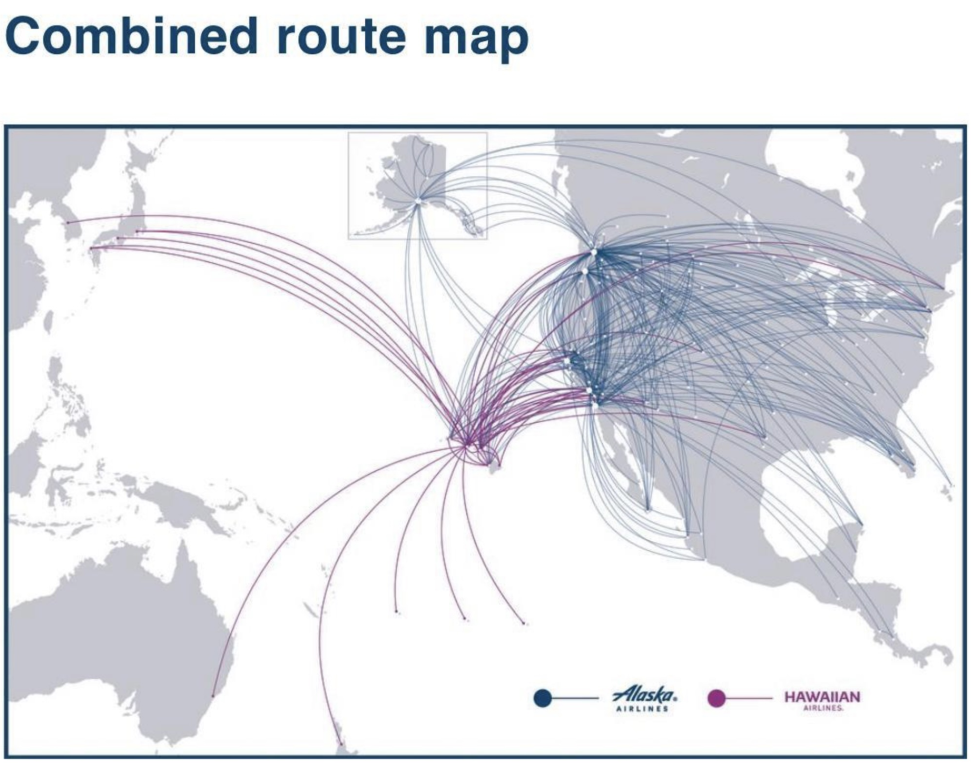
Air North Route Map



Alaska and Hawaii Airlines Combined Route Math



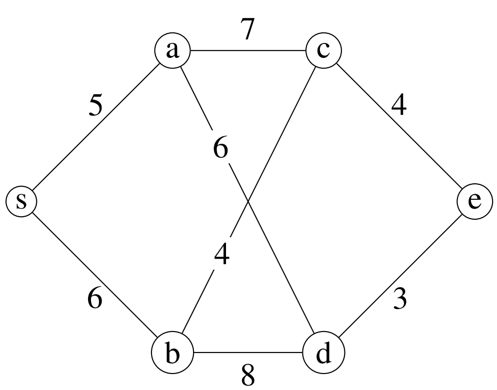
MATH F113X: Dijkstra's Algorithm

Dijkstra's Algorithm

input: a graph with distances (weights) on the edges, a starting vertex, say s and end ending vertex, say e
output: the length of the shortest path between s and e
rough strategy: Starting with the ending vertex, work your way back to the starting vertex keeping track of the shortest path **thus far.**
Steps:

1. Mark the ending vertex with a distance of zero and label it as **current**.
   (FYI: Once a **current** vertex is explored, it will be labelled **visited** and never considered again.)
2. Let v be the current vertex. For every vertex w with an edge to v **not marked as visited**, calculate the distance from w to e through v. If this distance is smaller than the present distance, update w with the new distance.** Otherwise, do nothing.
   (FYI: This number is called the tentative distance to e.)
3. Mark the current vertex as **visited.** Never look at this vertex again.
4. Identify the **un-visited** vertex with the smallest distance to e. Mark it as current and return to step 2. You know when to stop when vertex s is labeled as **current**.

** If you keep track of which current vertex updates a tentative minimum distance, you can recover the shortest path itself, not just the length.



vertex	current/ visited	**tentative** minimum distance to e	preceding vertex
s			
a			
b			
c			
d			
e			

Length of the shortest path from s to e: ____________

Find the shortest path from s to e **using the last column in the table.**

Think of another application of Dijkstra's Algorithm. It must include: vertices, weights of edges, and the meaning of a shortest path.