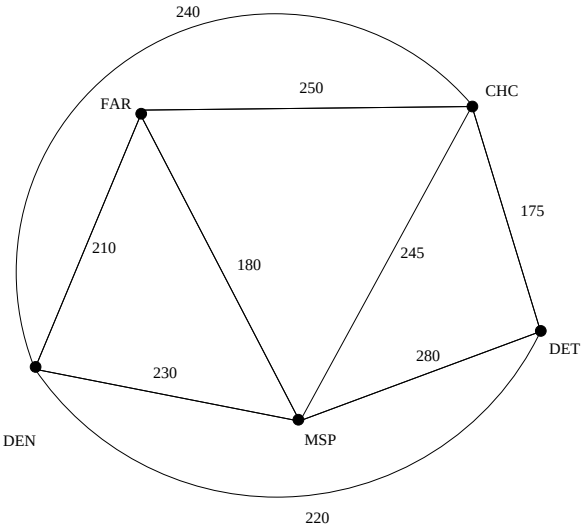
Math 310
Weighted Graphs and Shortest Path Problems Handout

Definitions:

- A **weighted graph** is a graph Γ with a number assigned to each edge of the graph. An assigned number is called the **weight** of the edge, and the collection of all weights is called a **weighting** of the graph Γ.
- Intuitively speaking, the weight of an edge represents the "cost" of traversing that edge. The specific meaning of this number depends on the situation being modeled by the graph. Weighted graphs can be used to represent applications from delivery routes for letter carriers to computer networks.
- In a weighted graph, the **length** of a simple path in the graph is the sum of the weights on each edge in the path. A **shortest** path between a pair of vertices in a graph is a path of minimal length among all possible paths between that pair of vertices.
- The **Traveling Salesman Problem** is to find a minimal length Hamiltonian Path representing the most efficient route that a traveling salesman can use to visit the finite number of cities on his route.
- A **shortest path algorithm** is a procedure that, when given a description of a finite graph and a pair of vertices in the graph (these vertices could be equal), finds a shortest path between the given pair of vertices.
- One shortest path algorithm is a **Brute Force Algorithm**, which finds a shortest path from u to v in s graph Γ by enumerating and computing the weight of every path between u and v one at a time and then outputs a minimal path. Notice that to solve the traveling salesman problem for a complete graph on n vertices, one would need to compute the weight of $(n-1)!$ different paths [this is inefficient when n is large].
- When a problem is difficult to solve algorithmically, rather than taking the time to find an optimal solution (the best possible path) one will often use a faster procedure to find a "good" solution (one that is close to the optimal solution) called an **approximation algorithm**.

Example: Solve the Traveling Salesman Problem for the following graph:



Dijkstra's Algorithm:

- **Input:** A weighted connected simple graph Γ with all weights positive.
- Γ has vertices $a = v_0, v_1, v_2, \cdots, v_n = z$.
- **for** $i := 1$ **to** n

  $L(v_i) := \infty$

  $L(a) := 0$

  $S := \emptyset$ [This initializes vertex labels so that the label of vertex a is 0 and all other vertices have label ∞, and S is the empty set.]
- **while** $z \notin S$

  begin

  $u :=$ a vertex not in S with $L(u)$ minimal

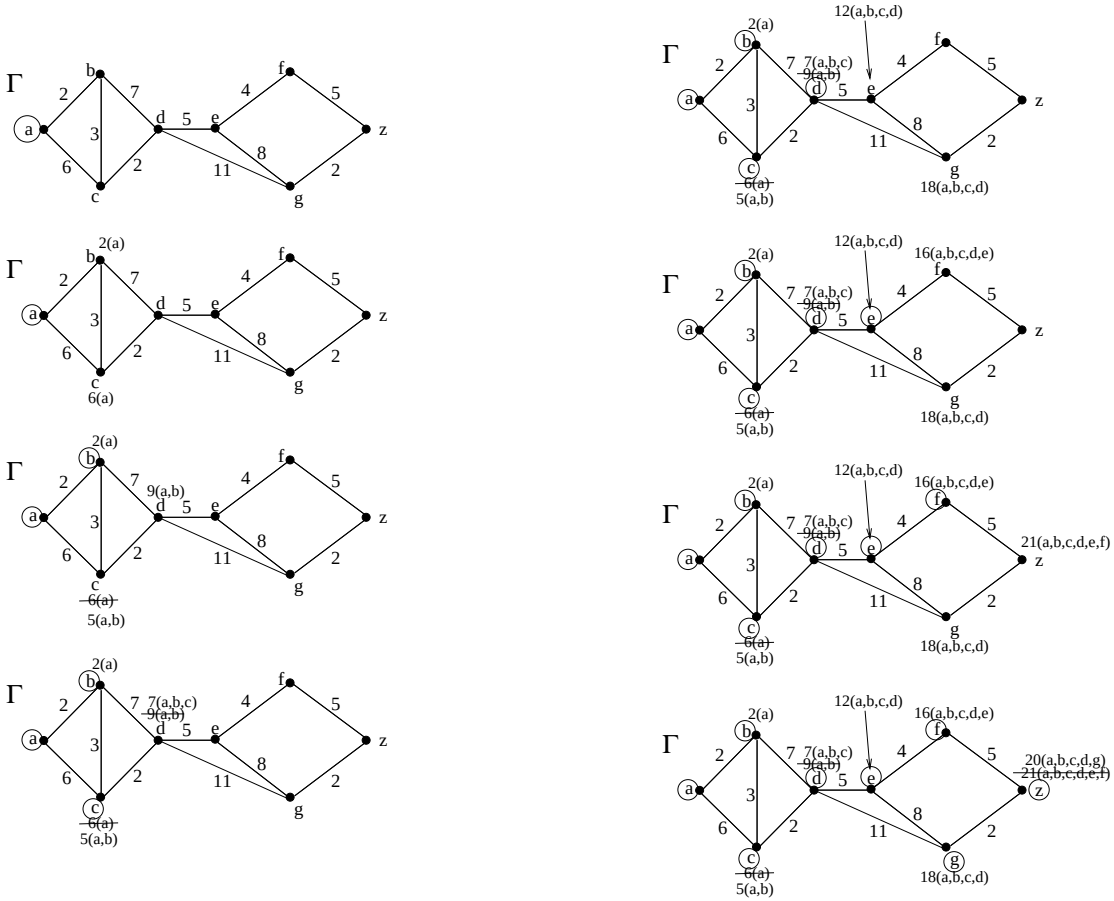
  $S := S \cup \{u\}$

  for all vertices $v \notin S$

  if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$ [This adds a vertex to S with minimal label and updates the labels of the vertices not in S].
- **end** $\{L(z) =$ length of a shortest path from a to $z\}$.

Notation: In this algorithm, (w, v) is the weight of the edge from u to w, and $L(u)$ is the length of a shortest path from a to u using only vertices in the set S.

Example: We will use Dijkstra's Algorithm to find a shortest path from a to z:



Therefore, the shortest path has degree sequence $\langle a, b, c, d, g, z \rangle$ and weight 20.