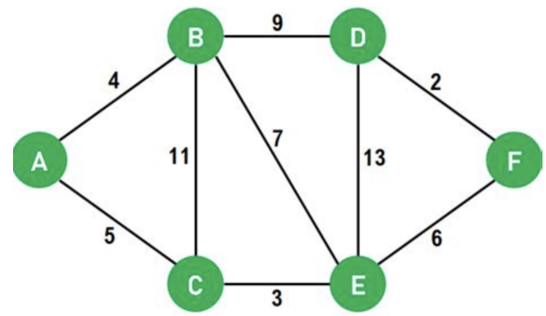
Last name ______________________

First name ______________________

LARSON—MATH 356—HOMEWORK WORKSHEET 02
Dijkstra's Algorithm.

1. What is a *shortest path* between two vertices in a weighted graph?



2. Find (eyeball) a shortest path from vertex F to vertex A in this graph.

Dijkstra's algorithm.
Let G be a graph with vertices V, edges E and edge weights $w((a,b))$ for each edge $(a,b) \in E$. Let $d(v,u)$ be the total weight of a shortest path from v to u. Initialize $S_0 = \{x\}$ and $U_0 = V \setminus \{x\}$.

After i iterations S_i will be the set of vertices where the shortest path from x to each of these vertices is known, and U_i will be the vertices where the shortest path from x has not yet been established. We'll always have $V = S_i \cup U_i$.

(a) Let x_i be a vertex that minimizes $d(x,s) + w((s,x_i))$ for $s \in S_i$, $x_i \in U_i$.

(b) Let $d(x,x_i)$ be the minimum found.

(c) Let $predecessor(x_i) = s$ where $s \in S_i$ is a vertex where $d(x,x_i) = d(x,s) + w((s,x_i))$.

(d) Let $S_{i+1} = S_i \cup \{x_i\}$.

(e) Let $U_{i+1} = U_i \setminus \{x_i\}$.

A shortest path from x to *any* vertex z (including y) can be found by using the *predecessor* function to trace backwards from z.

3. Now use Dijkstra's algorithm to find a shortest path from F to A.

(a) What are S_0 and U_0?

(b) Then there will be 5 iterations. So, you will first find x_1, S_1, U_1 and $predecessor(x_1)$.

(c) Then find x_2, S_2, U_2 and $predecessor(x_2)$. Repeat.

(d) Explain how to use the predecessor function to trace back a shortest path.

(e) Sum up the weights of the edges on this path to calculate $d(F,A)$.

Dijkstra's algorithm.

Let G be a graph with vertices V, edges E and edge weights $w((a,b))$ for each edge $(a,b) \in E$. Let $d(v,u)$ be the total weight of a shortest path from v to u. Initialize $S_0 = \{x\}$ and $U_0 = V \setminus \{x\}$.

After i iterations S_i will be the set of vertices where the shortest path from x to each of these vertices is known, and U_i will be the vertices where the shortest path from x has not yet been established. We'll always have $V = S_i \cup U_i$.

(a) Let x_i be a vertex that minimizes $d(x,s) + w((s,x_i))$ for $s \in S_i$, $x_i \in U_i$.

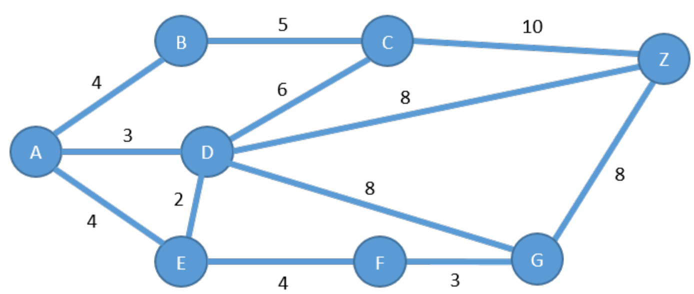
(b) Let $d(x,x_i)$ be the minimum found.

(c) Let $predecessor(x_i) = s$ where $s \in S_i$ is a vertex where $d(x,x_i) = d(x,s) + w((s,x_i))$.

(d) Let $S_{i+1} = S_i \cup \{x_i\}$.

(e) Let $U_{i+1} = U_i \setminus \{x_i\}$.

A shortest path from x to *any* vertex z (including y) can be found by using the *predecessor* function to trace backwards from z.



4. Find (eyeball) a shortest path from vertex Z to vertex A in this graph.

5. Now use Dijkstra's algorithm to find a shortest path from Z to A.

   (a) What are S_0 and U_0?

   (b) Then there will be 7 iterations. So, you will first find x_1, S_1, U_1 and $predecessor(x_1)$.

   (c) Then find x_2, S_2, U_2 and $predecessor(x_2)$. Repeat.

   (d) Explain how to use the predecessor function to trace back a shortest path.

   (e) Sum up the weights of the edges on this path to calculate $d(Z,A)$.