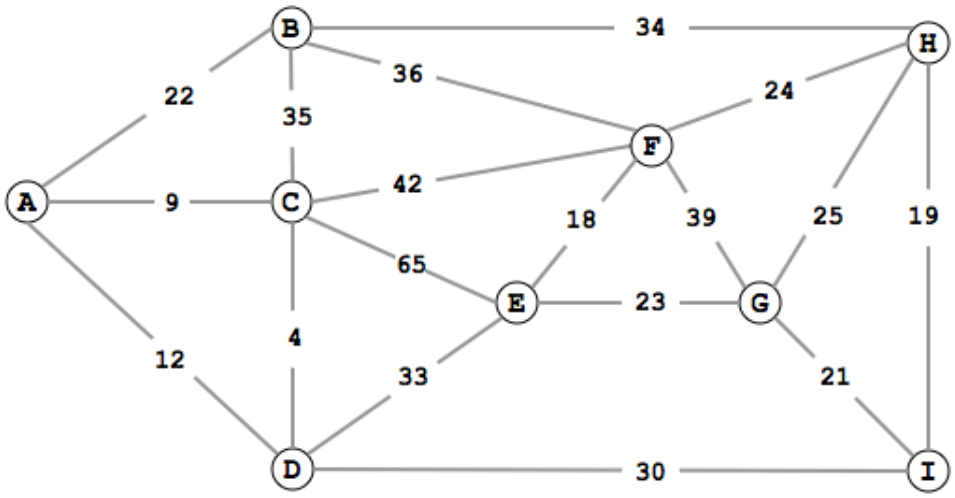
Last name _____

First name _____

LARSON—MATH 356–CLASSROOM WORKSHEET 31
Dijkstra's Shortest Path Algorithm.



1. Find a shortest weighted path from vertex D to vertex F.

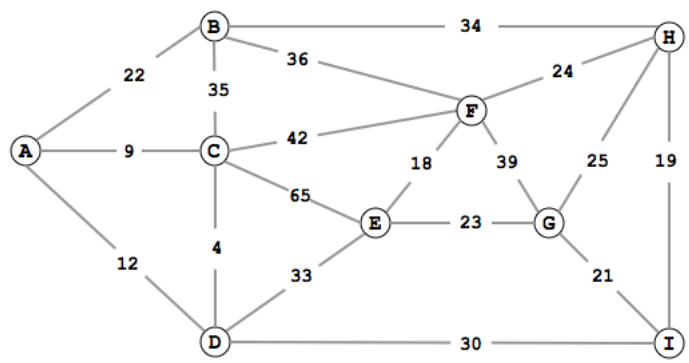
2. How many steps will our naive algorithm use to a shortest weighted path from vertex D to vertex F.

Algorithm 3.5 A general template for Dijkstra's algorithm.

Input An undirected or directed graph $G = (V, E)$ that is weighted and has no self-loops. The order of G is $n > 0$. A vertex $s \in V$ from which to start the search. Vertices are numbered from 1 to n, i.e. $V = \{1, 2, \ldots, n\}$.

Output A list D of distances such that $D[v]$ is the distance of a shortest path from s to v. A list P of vertex parents such that $P[v]$ is the parent of v, i.e. v is adjacent from $P[v]$.

```
D ← [∞, ∞, ..., ∞]                          ▷ n copies of ∞
D[s] ← 0
P ← []
Q ← V                                       ▷ list of nodes to visit
while length(Q) > 0 do
    find v ∈ Q such that D[v] is minimal
    Q ← remove(Q, v)
    for each u ∈ adj(v) ∩ Q do
        if D[u] > D[v] + w(vu) then
            D[u] ← D[v] + w(vu)
            P[u] ← v
return (D, P)
```



3. Use Dijkstra's Algorithm to find a shortest weighted path from vertex C to vertex H.