**Last name** _____

**First name** _____

# LARSON—MATH 556—CLASSROOM WORKSHEET 05
## Shortest Path Problem.

### Concepts & Notation

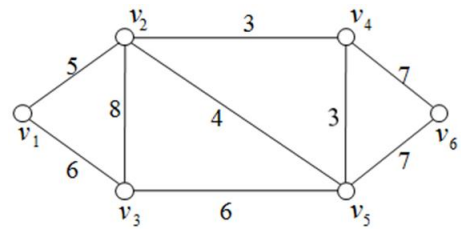
- Sec. 1.1: $G$, graph, $V(G)$, vertices, $E(G)$, edges, planar, incident, adjacent, loop/link, $\nu(G)$, $\epsilon(G)$
- Sec. 1.2: isomorphic, $G \cong H$, complete graph, $K_n$, empty graph, bipartite graph, complete bipartite graph $K_{m,n}$, $k$-partite graph, complete $k$-partite graph, $k$-cube, complement, $G^c$, $K_n^c$, $K_{m,n}^c$, automorphism, vertex-transitive, edge-transitive.
- Sec. 1.3: incidence matrix, $\mathbf{M}(G)$, adjacency matrix, $\mathbf{A}(G)$.
- Sec. 1.4: subgraph, spanning subgraph, induced subgraph, $G[V']$, edge-induced subgraph, $G[E']$, union $G_1 \cup G_2$, $G_1 + G_2$, intersection $G_1 \cap G_2$.
- Sec. 1.5: degree, $k$-regular graph, regular graph, degree sequence, graphic; $1^{st}$ Theorem.
- Sec. 1.6: walk, length, trail, path, components, connected, distance, $d_G(u, v)$, diameter
- Sec. 1.7: closed, cycle, girth; Bipartite Characterization Theorem
- Sec. 1.8: $w(e)$, shortest path problem, tree; Dijkstra's algorithm.
- Sec. 2.1: acyclic; 3 tree theorems.

### Basic Shortest Path Algorithm

The idea of the basic shortest path algorithm to find a shortest path from vertex $u_0$ to vertex $v_0$ in a graph is to maintain a set $S$ including $v$ where the shortest distance from $u_0$ to each vertex in $S$ is known, and to extend $S$ by a single vertex in $\bar{S}$ at each step, using the following formula:

$$d(u_0, \bar{S}) = \min_{u \in S, v \in \bar{S}} \{d(u_0, u) + w(uv)\}$$

Given the distances to all the vertices in $S$ you then consider each of the edges $uv$ from $S$ to $\bar{S}$, find a minimizing vertex $u$, add it to $S$ and iterate. Eventually $S$ must equal the entire vertex set.




1. Use the Basic Shortest Path Algorithm to find a shortest path from $v_1$ to $v_6$. Include your steps.

**Dijkstra's Shortest Path Algorithm**

The main idea is similar. It is a modification designed to keep track of intermediate computations and only consider the edges from *one* vertex in $\bar{S}$ at each step. It is called a *labeling algorithm* because at each step we maintain a label $l(v)$ for each vertex $v$ that stores any intermediate computations.

1. Set $l(u_0) = 0$, $l(v) = \infty$ for $v \neq u_0$, $S_0 = \{u_0\}$ and $i = 0$.
2. For each $v \in \bar{S}_i$, replace $l(v)$ by $\min\{l(v), l(u_i) + w(u_i v)\}$. Compute $\min_{v \in \bar{S}_i}\{l(v)\}$ and let $u_{i+1}$ denote a vertex for which this minimum is attained. Set $S_{i+1} = S_i \cup \{u_{i+1}\}$.
3. If $i = \nu - 1$, stop. If $i < \nu - 1$, replace $i$ by $i + 1$ and go to step 2.

2. Use Dijkstra's Algorithm to find a shortest path from $v_1$ to $v_6$. Include your steps.