Instance variables store algorithm data.
marked[v] is true iff v connected to s.
edgeTo[v] is vertex visited to get to v.
BreadthFirstPaths constructor computes the result of the algorithm with the bfs iterative method.

Cost model given undirected graph?
Each vertex is visited at most once.
Each edge is checked at most twice.
Dijkstra's Pseudocode

PQ.add(s, 0)

For all other vertices v, PQ.add(v, infinity)

While PQ is not empty:
  p = PQ.removeSmallest()
  Relax all edges from p

Relaxing an edge (v, w) with weight:
If distTo[w] > distTo[v] + weight:
  distTo[w] = distTo[v] + weight
  edgeTo[w] = v
  PQ.changePriority(w, distTo[w])

Invariants
edgeTo[v]: best known predecessor of v.
distTo[v]: best known distance of s to v.
PQ maintains vertices based on distTo.

Important properties
Always visits vertices in order of total distance from source. Relaxation always fails on edges to visited (white) vertices.

Dijkstra's Algorithm Correctness

Dijkstra's algorithm. Visit vertices in order of distance from source.
On visit, relax every edge from the visited vertex.

Dijkstra's can fail if the graph has negative weight edges. Give an example graph.

A* Search Algorithm

Dijkstra's algorithm with one modification.
• Dijkstra's algorithm: Priority is defined by distTo[v] only.
• A* search: Priority is defined by distTo[v] + h(v, goal).

Where h(v, goal) is a heuristic: an estimate of the distance from v to the goal.

Computing a Heuristic

Where h(v, goal) is a heuristic: an estimate of the distance from v to the goal.
For maps, we can use Euclidean distance (right triangle hypotenuse length).

Will A* search return the correct shortest path if h(v, goal) = 10 for every v in the graph?