Code for Dijkstra's Algorithm

```
/**
 * Return the index of the smallest element of distances,
 * ignoring those in visited.
 */
protected int cheapest(double[] distances, boolean[] visited) {
  int best = -1;
  for (int i = 0; i < size(); i++) {
    if (!visited[i]
        && ((best < 0) || (distances[i] < distances[best]))) {
      best = i;
    }
  }
  return best;
}
```

```
/**
 * Return an array of the distances from source to each other
 * vertex.
 */
public double[] distancesFrom(int source) {
  double[] result = new double[size()];
  java.util.Arrays.fill(result, Double.POSITIVE_INFINITY);
  result[source] = 0;
  boolean[] visited = new boolean[size()];
  for (int i = 0; i < size(); i++) {
    int vertex = cheapest(result, visited);
    visited[vertex] = true;
    for (int j = 0; j < size(); j++) {
      result[j] = Math.min(result[j],
                           result[vertex] + getCost(vertex, j));
    }
  }
  return result;
}
```