

parseIntersection(line: String, intersections: Map < String, Intersection >; directedEdges: Map < DirectedEdge, DirectedEdgeInfo >; coordChecker: List < Coord >; referencesMadeByIntersections: HashSet < String >)	
String[] parts = line.split("\\s+");	
String name = parts[0];	
validateIntersectionInput(name, parts, intersections, line)	
double x = checkCoordinateComponent(parts[1]);	
double y = checkCoordinateComponent(parts[2]);	
List < String > dests = new ArrayList();	
List < Double > probs = new ArrayList();	
for i = 3 to parts.length - 1 by 2	
String dest = parts[i];	
double relProb = checkProbability(parts[i + 1]);	
dests.add(dest)	
probs.add(relProb)	
DirectedEdge forward = new DirectedEdge(name, dest)	
DirectedEdge reverse = new DirectedEdge(dest, name)	
directedEdges.putIfAbsent(forward, new DirectedEdgeInfo())	
directedEdges.putIfAbsent(reverse, new DirectedEdgeInfo())	
processIntersection(name, x, y, probs, dests, coordChecker, referencesMadeByIntersections, intersections)	