# Prim's Algorithm

**procedure** MST(Graph *graph*)

Map *dist* = a Map from vertex to distance (number), initially empty
Map *parent* = a Map from vertex to vertex, initially empty
Set *visited* = a Set of vertices, initially empty
Graph *tree* = an empty graph

*start* = pick arbitrary vertex
Initialize *parent* and *dist* with vertices adjacent to *start*

**while** $visited.size() < graph.size()$ **do**
*current* = find the closest, unvisited vertex in *dist*
$visited.add(current)$

$tree.addEdge(current, parent(current))$

**for** *other* in $graph.adjacent(current)$ and *other* not in *visited* **do**
$newDist = graph.edgeWeight(current, other)$

**if** $newDist < dist(other)$ or *other* not in *dist* **then**
$dist.push(other, newDist)$
$parent.push(other, current)$
**end if**

**end for**

**end while**

**return** *tree*

**end procedure**