

dijkstra(start, target)

path = new dictionary (location to prev location)

distance = new dictionary (location to distance)

workList = new minPriorityQueue();

first PQ element = new Long[] {start.id, null};

workList.enqueue(first PQ element, 0.0);

path.put(start, null)

distance.put(start, 0);

while workList.size() > 0

PQElement<Long[]> curr = workList.dequeue()

for neighbors of curr

if not a building

if not in distance

workList.enqueue([neighbor, curr.data[0]], distance)

Long[] nextTemp = workList.peek()

Long next = nextTemp.data[0]

if distance of nextTemp < distance.get(next)

distance.put(next, nextTemp.priority)

path.put(next, nextTemp.data[1])

workList.dequeue()

// making return value

result = new ArrayList();

result.add(target)

loc = path.get(target)

while (loc != start)

! result.addFront(loc)

! loc = path.get(loc)

result.addFront(loc)

return result