Algorithm 1 RRT Algorithm

Input: Map $\mathcal{M}$, start point x_{init}, goal point x_{goal}
Output: A path $\mathcal{P}$ from x_{init} to x_{goal}

$\mathcal{T}$.init()
for $i = 1$ to n **do**
    $x_{rand} \leftarrow$ Sample($\mathcal{M}$)
    $x_{near} \leftarrow$ NearestNeighbor(x_{rand},$\mathcal{T}$)
    $x_{new} \leftarrow$ Step(x_{rand}, x_{near}, STEP_SIZE)
    if CollisionFree(x_{new}, x_{near}) **then**
        $\mathcal{T}$.AddNode(x_{new})
        $\mathcal{T}$.AddEdge(x_{near}, x_{new})
    if Distance(x_{new}, x_{goal}) < STEP_SIZE **then**
        Break
return $\mathcal{T}$

Algorithm 2 Goal-Biased RRT Algorithm

Input: Map $\mathcal{M}$, start point x_{init}, goal point x_{goal}
Output: A path $\mathcal{P}$ from x_{init} to x_{goal}

$\mathcal{T}$.init()
for $i = 1$ to n **do**
    $x_{rand} \leftarrow$ Sample($\mathcal{M}$, x_{goal}, EXPLORE_RATE)
    $x_{near} \leftarrow$ NearestNeighbor(x_{rand},$\mathcal{T}$)
    $x_{new} \leftarrow$ Step(x_{rand}, x_{near}, STEP_SIZE)
    if CollisionFree(x_{new}, x_{near}) **then**
        $\mathcal{T}$.AddNode(x_{new})
        $\mathcal{T}$.AddEdge(x_{near}, x_{new})
    if Distance(x_{new}, x_{goal}) < STEP_SIZE **then**
        Break
return $\mathcal{T}$