Algorithm 2 Fast Marching Tree Algorithm (FMT*): Details

$V \leftarrow \{x_{\text{init}}\} \cup \texttt{SampleFree}(n)$; $E \leftarrow \emptyset$
$V_{\text{unvisited}} \leftarrow V \backslash \{x_{\text{init}}\}$; $V_{\text{open}} \leftarrow \{x_{\text{init}}\}$, $V_{\text{closed}} \leftarrow \emptyset$
$z \leftarrow x_{\text{init}}$
$N_z \leftarrow \texttt{Near}(V \backslash \{z\}, z, r_n)$
$\texttt{Save}(N_z, z)$
while $z \notin \mathcal{X}_{\text{goal}}$ **do**
  $V_{\text{open, new}} \leftarrow \emptyset$
  $X_{\text{near}} = N_z \cap V_{\text{unvisited}}$
  for $x \in X_{\text{near}}$ **do**
    $N_x \leftarrow \texttt{Near}(V \backslash \{x\}, x, r_n)$
    $\texttt{Save}(N_x, x)$
    $Y_{\text{near}} \leftarrow N_x \cap V_{\text{open}})$
    $y_{\min} \leftarrow \arg\min_{y \in Y_{\text{near}}} \{c(y) + \texttt{Cost}(y, x)\}$ // dynamic programming equation
    if $\texttt{CollisionFree}(y_{\min}, x)$ **then**
      $E \leftarrow E \cup \{(y_{\min}, x)\}$ // straight line joining $y_{\min}$ and x is collision-free
      $V_{\text{open, new}} \leftarrow V_{\text{open, new}} \cup \{x\}$
      $V_{\text{unvisited}} \leftarrow V_{\text{unvisited}} \backslash \{x\}$
      $c(x) = c(y_{\min}) + \texttt{Cost}(y_{\min}, x)$ // cost-to-arrive from x_{init} in tree $T = (V_{\text{open}} \cup V_{\text{closed}}, E)$
    end if
  end for
  $V_{\text{open}} \leftarrow (V_{\text{open}} \cup V_{\text{open, new}}) \backslash \{z\}$
  $V_{\text{closed}} \leftarrow V_{\text{closed}} \cup \{z\}$
  if $V_{\text{open}} = \emptyset$ **then**
    return Failure
  end if
  $z \leftarrow \arg\min_{y \in V_{\text{open}}} \{c(y)\}$
end while
return $\texttt{Path}(z, T = (V_{\text{open}} \cup V_{\text{closed}}, E))$