Algorithm 1 Q-λ learning

```
function QLambdaLearning($\alpha, \gamma, \lambda$)
    Initialize $\theta$, hidden state $x_0$ and belief state $b_0$
    $t \leftarrow 0$
    $a_0, a_0^* \leftarrow$ ChooseAction($\theta, b_0$)
    $\xi_0 \leftarrow$ GridWeights($b_0, a_0^*$)
    $e \leftarrow \mathbf{0}$
    loop
        Observe reward $r_t$ and observation $y_{t+1}$
        $b_{t+1} \leftarrow$ UpdateBeliefState($b_t, a_t, y_{t+1}$)
        $a_t, a_t^* \leftarrow$ ChooseAction($\theta, b_t$)
        if $a_t \neq a_t^*$ then
            $e \leftarrow \mathbf{0} + \xi_t$
        else
            $e \leftarrow e + \xi_t$
        $\xi_{t+1} \leftarrow$ GridWeights($b_t, a_t$)
        $\xi_{t+1}^* \leftarrow$ GridWeights($b_t, a_t^*$)
        $\delta \leftarrow r_t + \gamma \cdot \theta^T \xi_{t+1}^* - \theta^T \xi_t$
        $\theta \leftarrow \theta + \alpha \cdot \delta \cdot e$
        $e \leftarrow \gamma \cdot \lambda \cdot e$
        $t \leftarrow t + 1$

function ChooseAction($\theta, b$)                                ▷ $\epsilon$-greedy
    $a^* \leftarrow \arg\max_a \theta^T$[GridWeights($b, a$)]
    if Uniform(0,1) $\leq \epsilon$ then
        $a \leftarrow$ random action
    else
        $a \leftarrow a^*$
    return $a, a^*$

function GridWeights($b, a$)
    $\xi \leftarrow \mathbf{0}$
    for each particle $p \in b$ do
        $\xi_a \leftarrow \xi_a +$ Interpolants($p$)
    $\xi \leftarrow \frac{\xi}{\sum_{a,i}(\xi_{ai})}$
    $\xi_{a\ i+1} \leftarrow \mathrm{var}(\xi)$
    return $\xi$
```

Algorithm 2 Modified Upper Confidence Bound for Trees

```
function SelectAction($(b, d)$)
    Initialize $Q$ and $N$
    $s \leftarrow$ random state from belief state $b$
    loop
        Simulate($(s, d)$)
    return $\arg\max_a Q(a)$

function Simulate($(s, d)$)
    if $d = 0$ then
        return $0$
    if $S \notin T$ then
        for $a \in A(s)$ do
            $(N(a), Q(a)) \leftarrow (N_0(a), Q_0(a))$
        $T = T \cup \{s\}$
        return Rollout($(s, d)$)
    $a \leftarrow$ ChooseAction($(s)$)
    $(s', r) \sim G(s, a)$
    $q \leftarrow r + \gamma$Simulate$(s', d - 1)$
    $N(a) \leftarrow N(a) + 1$
    $Q(a) \leftarrow Q(a) + \frac{q - Q(a)}{N(a)}$

function Rollout($(s, d)$)
    if $d = 0$ then
        return $0$
    $a \leftarrow$ random action $a \in A$
    $(s', r) \sim G(s, a)$
    return $r + \gamma$Rollout$(s', d - 1)$

function ChooseAction($(s)$)
    for $a \in A$ do
        if $N(s, a) = 0$ then return a
    return $\arg\max_a Q(a) + c\sqrt{\frac{\log(\sum_a N(a))}{N(a)}}$
```