n-step Tree Backup for estimating $Q \approx q_*$ or q_π

Initialize $Q(s, a)$ arbitrarily, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize π to be greedy with respect to Q, or as a fixed given policy
Algorithm parameters: step size $\alpha \in (0, 1]$, a positive integer n
All store and access operations can take their index mod $n + 1$

Loop for each episode:
    Initialize and store $S_0 \neq$ terminal
    Choose an action A_0 arbitrarily as a function of S_0; Store A_0
    $T \leftarrow \infty$
    Loop for $t = 0, 1, 2, \ldots$:
    | If $t < T$:
    | Take action A_t; observe and store the next reward and state as R_{t+1}, S_{t+1}
    | If S_{t+1} is terminal:
    | $T \leftarrow t + 1$
    | else:
    | Choose an action A_{t+1} arbitrarily as a function of S_{t+1}; Store A_{t+1}
    | $\tau \leftarrow t + 1 - n$ (τ is the time whose estimate is being updated)
    | If $\tau \geq 0$:
    | If $t + 1 \geq T$:
    | $G \leftarrow R_T$
    | else
    | $G \leftarrow R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a)$
    | Loop for $k = \min(t, T - 1)$ down through $\tau + 1$:
    | $G \leftarrow R_k + \gamma \sum_{a \neq A_k} \pi(a|S_k) Q(S_k, a) + \gamma \pi(A_k|S_k) G$
    | $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha \left[G - Q(S_\tau, A_\tau)\right]$
    | If π is being learned, then ensure that $\pi(\cdot|S_\tau)$ is greedy wrt Q
    Until $\tau = T - 1$