

# Monte Carlo Tree Search

---

## Algorithm 1 Monte Carlo Tree Search

---

```

1: function MCTS( $s_{root}$ )
2:   while within computational budget do
3:      $s \leftarrow \text{TreePolicy}(s_{root})$ 
4:      $\text{winner} \leftarrow \text{DefaultPolicy}(s)$ 
5:     Backup( $s$ ,  $\text{winner}$ )
6:   end while
7:   return Action( $\text{argmax}_{s' \in \text{children}(s_{root})} \frac{Q(s')}{N(s')}$ )
8: end function
9: function TREEPOLICY( $s$ )
10:  while  $s$  is not terminal do
11:    if  $s$  is not fully expanded then
12:      return Expand( $s$ ) ▷  $c$  is a hyperparameter
13:    else
14:       $s \leftarrow \text{BestChild}(s, c)$ 
15:    end if
16:  end while
17:  return  $s$ 
18: end function
19: function EXPAND( $s$ )
20:   $\text{child} \leftarrow$  previously unexpanded child of  $s$ 
21:  Update the tree with  $(s, \text{child})$ 
22:  return  $\text{child}$ 
23: end function
24: function BESTCHILD( $s, c$ )
25:  return  $\text{argmax}_{s' \in \text{children}(s)} \left( \frac{Q(s')}{N(s')} + c \sqrt{\frac{2 \ln N(s)}{N(s')}} \right)$ 
26: end function
27: function DEFAULTPOLICY( $s$ )
28:  while  $s$  is not terminal do
29:     $s \leftarrow$  random child of  $s$ 
30:  end while
31:  return  $\text{winner}$ 
32: end function
33: function BACKUP( $s$ ,  $\text{winner}$ )
34:  while  $s$  is not Null do ▷ Parent of root is Null
35:     $N(s) \leftarrow N(s) + 1$ 
36:     $Q(s) \leftarrow Q(s) + \Delta(s, \text{winner})$  ▷ Update winning count based on who's in control of  $s$ 
37:     $s \leftarrow \text{parent}(s)$ 
38:  end while
39: end function

```

---

In the simplest form,  $\Delta(s, \text{winner})$  is 1 if the player **in control of**  $s$  (i.e.,  $s$  is the outcome of that player's action) has won in the rollout, and 0 otherwise.