mcts example

```
def monte_carlo_tree_search(self):
        if self.winning_move is not None:
            return self.winning_move

        end_time = time.time() + MCTS_TIME
        while time.time() < end_time:

            for child in self.root_node.children:  # winner found?
                if child.results == float('inf'):
                    self.root_node.show_data()
                    return child.move

            leaf = self.traverse_and_expand(self.root_node)  # traverse
            result = leaf.rollout()  # rollout
            leaf.backpropagate(result)  # backpropagate

        self.root_node.show_data()
        return self.get_best_move()
```

```
def traverse_and_expand(self, node: TreeNode0):
        while not node.is_leaf:
            node = self.best_uct(node)
        if len(node.moves) > 0 and node.sims > 0:
            node.expand_node()
            node = random.choice(node.children)
        return node
```

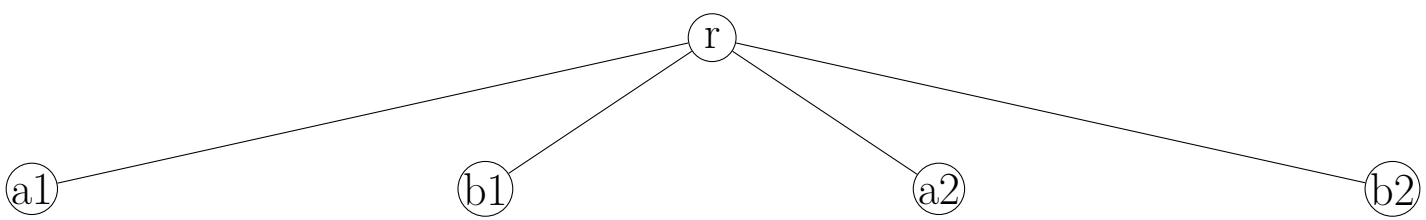
```
github repo mcts/main.py

= size 2          node names     5   6        a1  b1
= mcts x                           9  10        a2  b2

root node init * > 5, 6, 9, 10, done
```

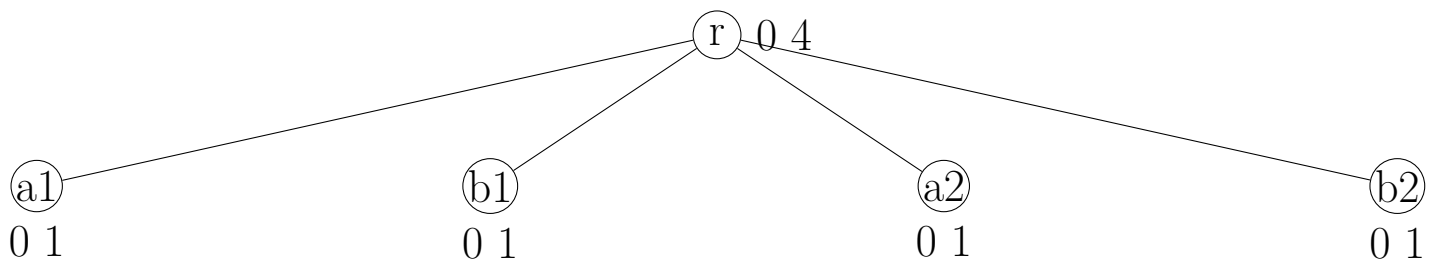
- **init tree, traverse**

- **expand**



```
trav_expand   best_uct 5 no_sims_yet
trav_expand leaf *  5
sim  1.    *  5 roll  6 10  9 parent loss
trav_expand   best_uct 6 no_sims_yet
trav_expand leaf *  6
sim  2.    *  6 roll 10  5  9 parent loss
trav_expand   best_uct 9 no_sims_yet
trav_expand leaf *  9
sim  3.    *  9 roll  5 10  6 parent loss
trav_expand   best_uct 10 no_sims_yet
trav_expand leaf * 10
sim  4.    * 10 roll  5  9  6 parent loss
```

- **traverse-expand, sims 1-4.**

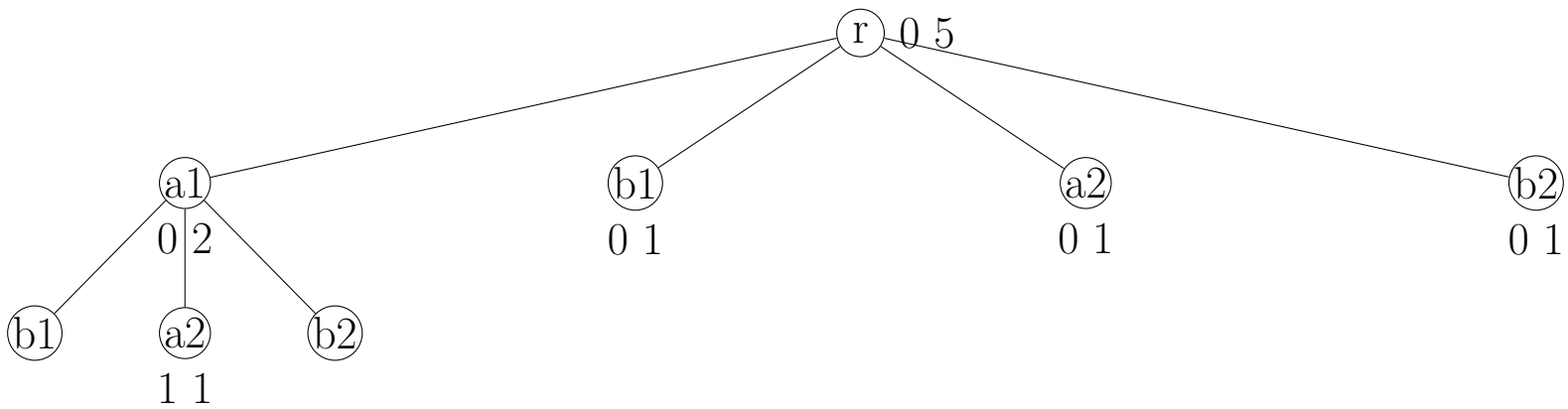


root node: win, visit(sim) counts: sum, over children

non-root: win, visit(sim) counts: for player-who-moved

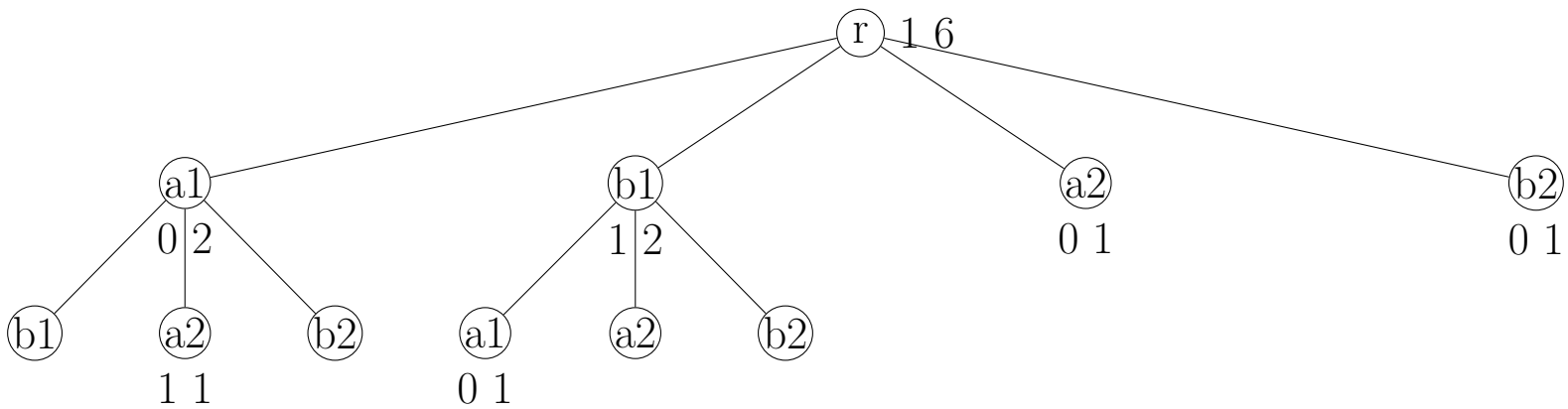
```
trav_expand   best_uct * 5
expand *  5 > 6
expand *  5 > 9
expand *  5 > 10
trav_expand leaf *  5  9
sim  5.    *  5  9 roll  6 10 parent win
```

- **traverse-expand, sim 5.**



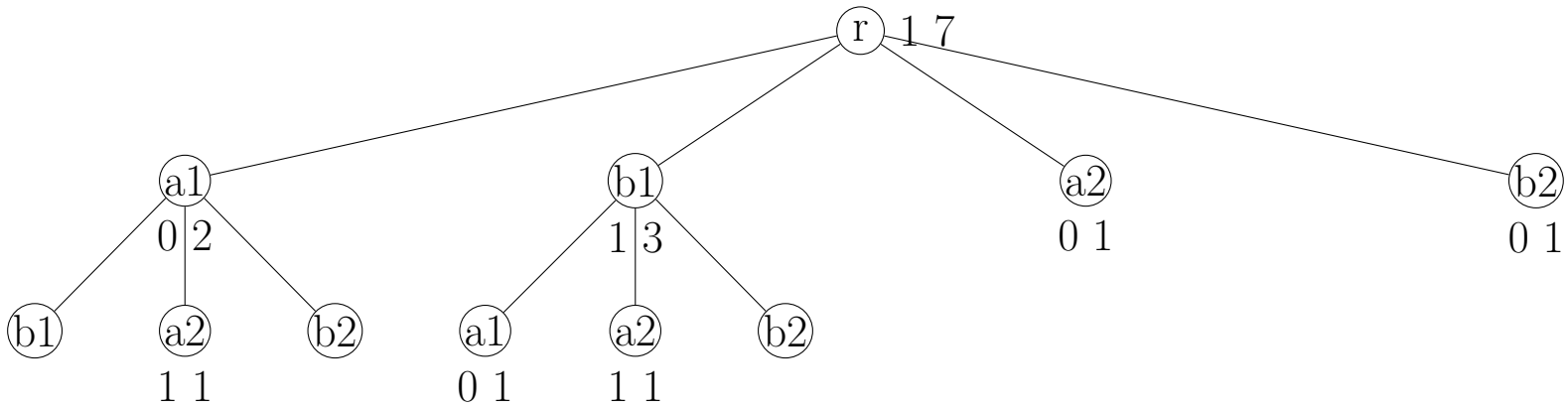
```
trav_expand   best_uct * 6
  expand *  6 > 5
  expand *  6 > 9
  expand *  6 > 10
  trav_expand leaf *  6  5
  sim  6.    *  6  5 roll 10 parent loss
```

- **traverse-expand, sim 6.**



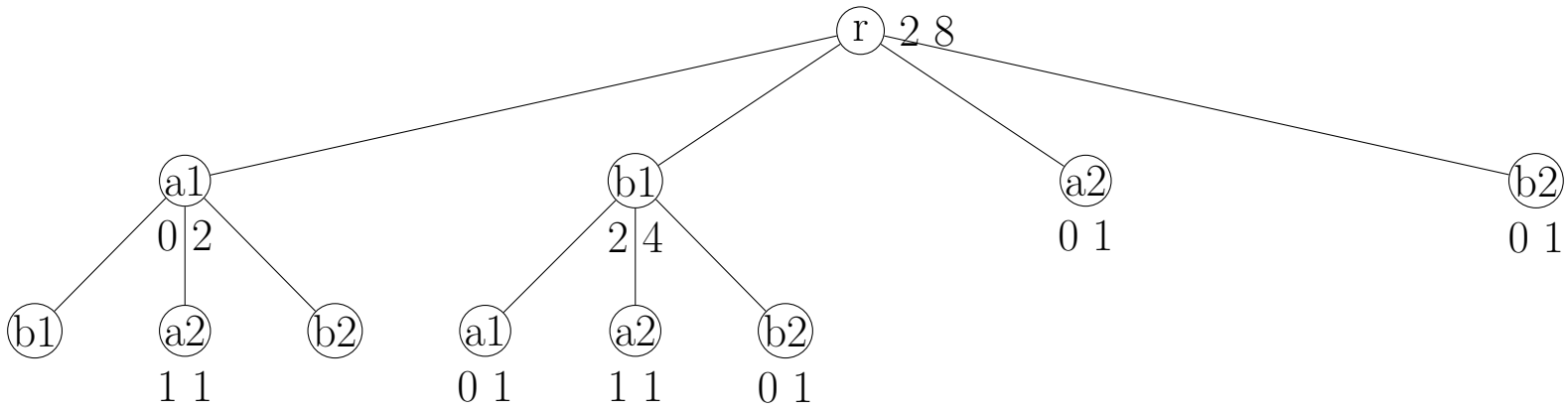
```
trav_expand   best_uct * 6  best_uct 9 no_sims_yet
  trav_expand leaf *  6  9
  sim  7.    *  6  9 roll  5 10 parent win
```

- **traverse-expand, sim 7.**



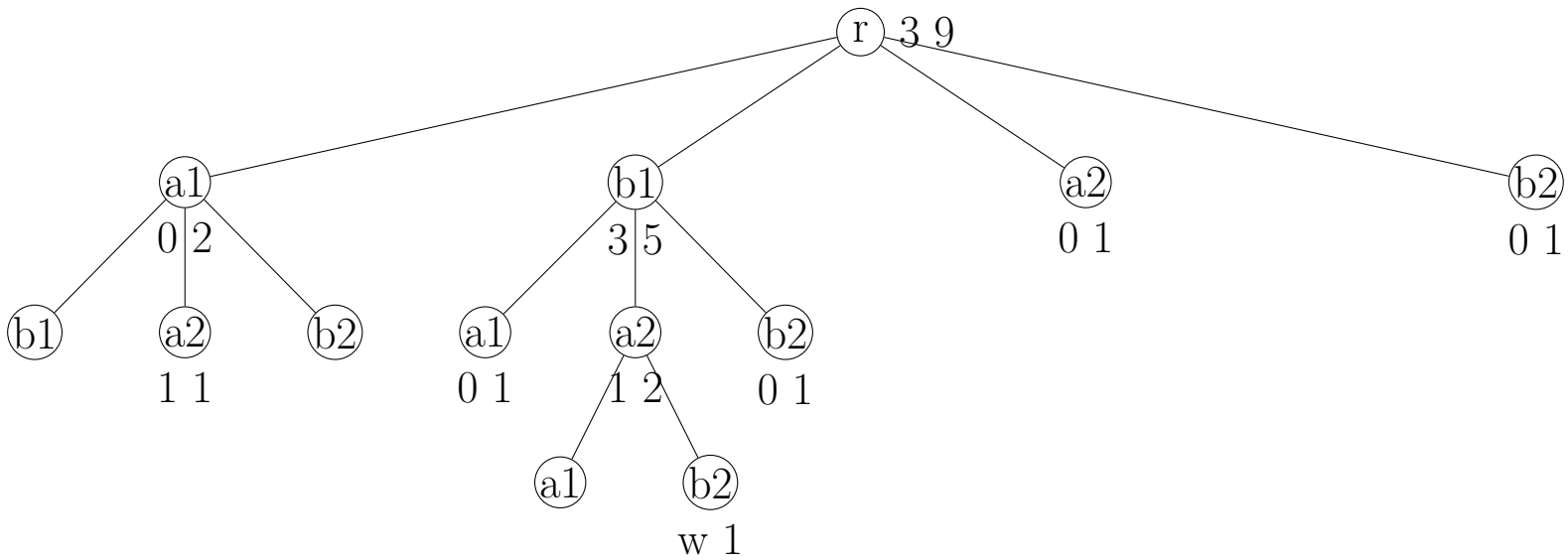
```
trav_expand   best_uct * 6  best_uct 10 no_sims_yet
trav_expand leaf *  6 10
sim  8.    *  6 10 roll  9 parent loss
```

- **traverse-expand, sim 8.**



```
trav_expand   best_uct * 6  best_uct 6 9
  expand *  6  9 > 5
  expand *  6  9 > 10
  sim  9. *  6  9 10 win
```

- **traverse-expand, sim 9.**



- **results after all 19 simulations**

move	sims	wins
5	1	0
6	1	0
9	11	inf
10	6	3
total	19	inf

```
   a b
 1  . . o
  2  x . o
      x x
```