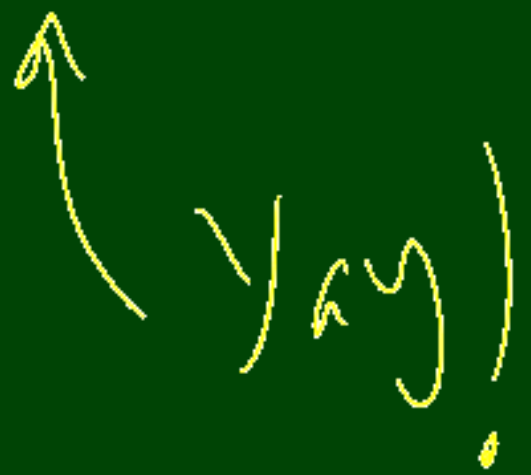
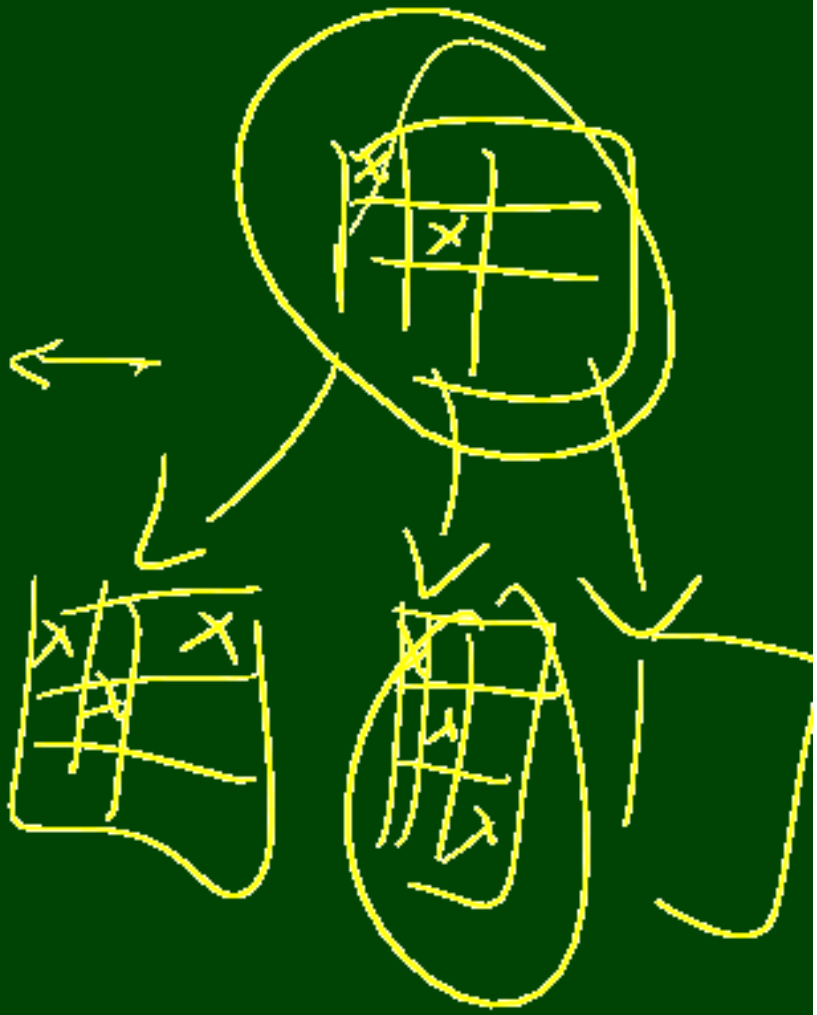
P3



Recursion To The Rescue


```
1  boolean win(Board b) {
2      if (b.threeXs()) {
3          return true;
4      }
5      else {
6          for (Move m : every possible move) {
7              if (win(b.do(move))) {
8                  return true;
9              }
10         }
11         return false;
12     }
```

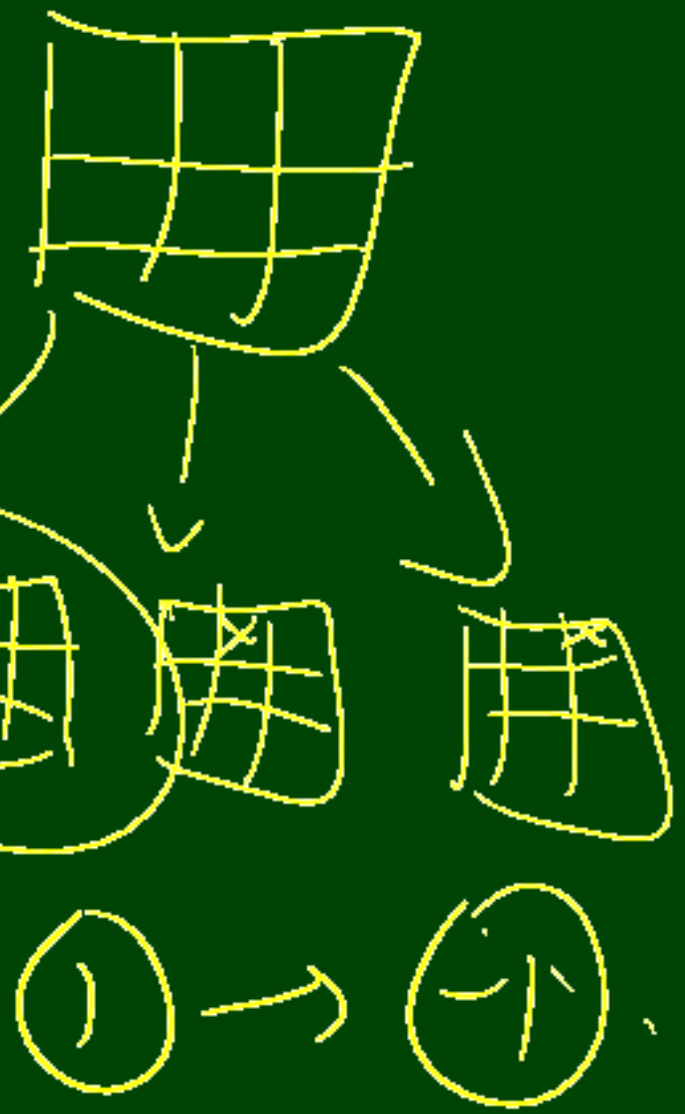


```
1  boolean win(Board b) {
2     if (b.threeXs()) {
3        return true;
4     }
5     else {
6        for (Move m : every possible move) {
7           if (win(b.do(move))) {
8              return true;
9           }
10       }
11       return false;
12    }
```

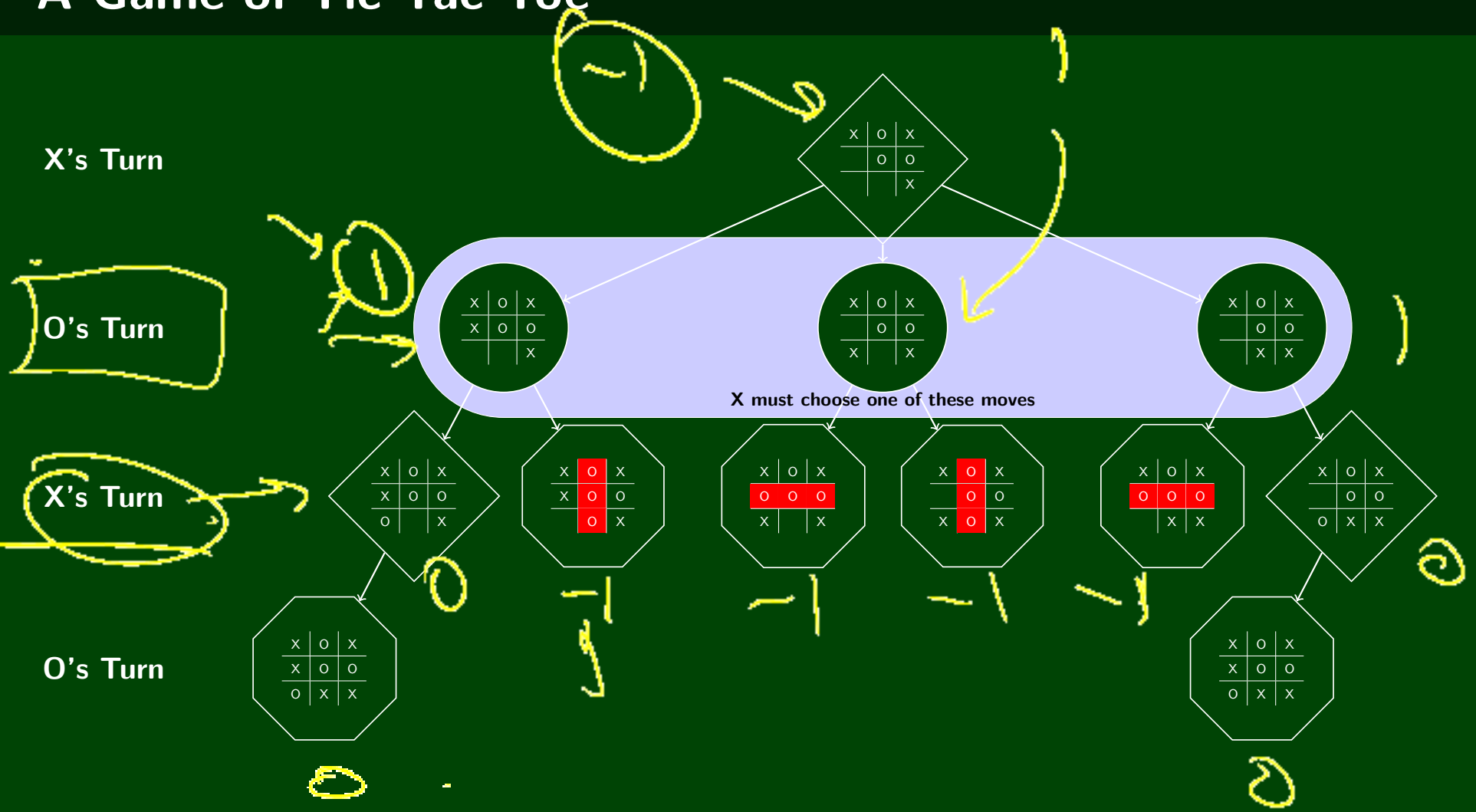
There's An Issue Here!

- When we make a move, it's not our turn any more.
- So the recursive call should be to **our opponent's option**
- Key Insight: Instead of guessing what the opponent is going to do, **assume she plays optimally**!

```
1  // +1 is a win; +0 is a draw; −1 is a loss
2  int eval(Board b) {
3     if (b.gameOver()) {
4        if (b.hasThree(me)) {
5           return 1;
6        }
7        else if (b.hasThree(them)) {
8           return −1;
9        }
10       else {
11          return 0;
12       }
13    }
14    else {
15       int best = −1;
16       for (Move m : every possible move) {
17          best = max(best, −eval(b.apply(move)));
18       }
19       return best;
20    }
```



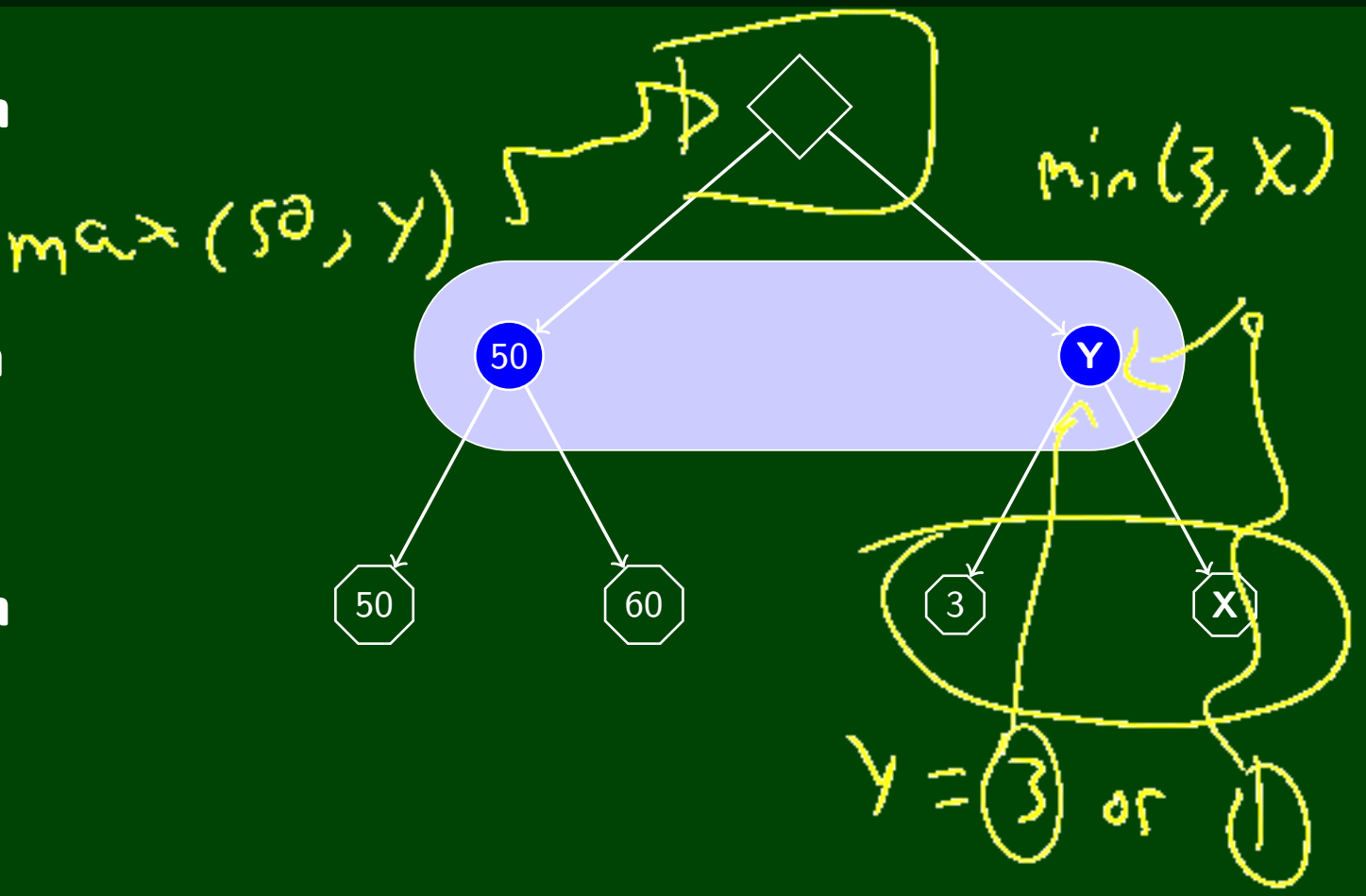
A Game of Tic-Tac-Toe



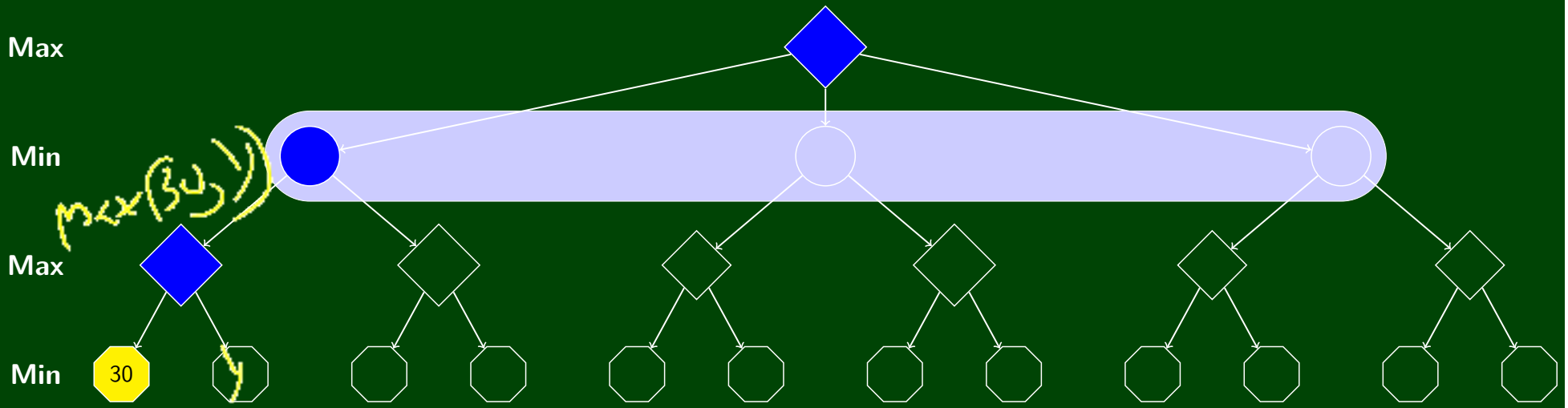
Max's Turn

Min's Turn

Max's Turn



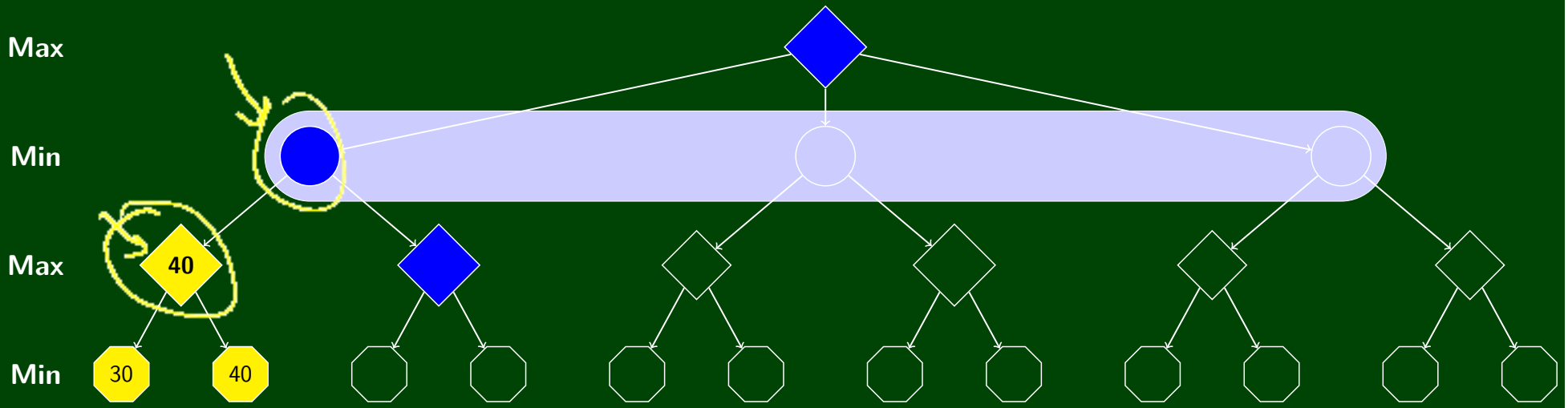
Pruning



Do we check the next node?

We currently have no information. So, yes!

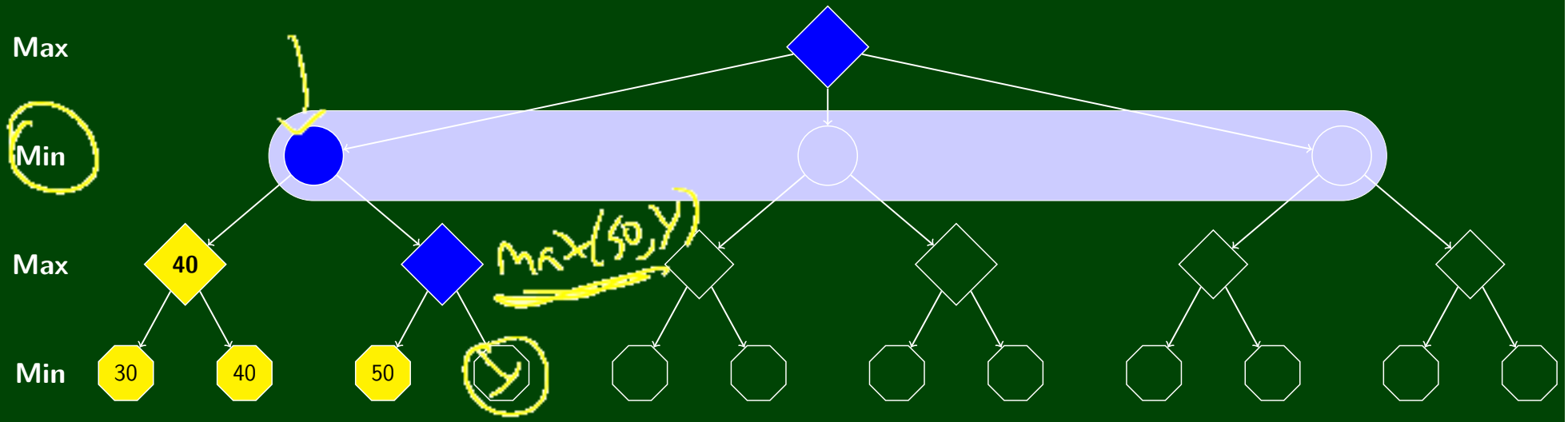
Pruning



Do we check the next node?

The current bounds are $[-\infty, 40]$. So, we **might** do better!

Pruning

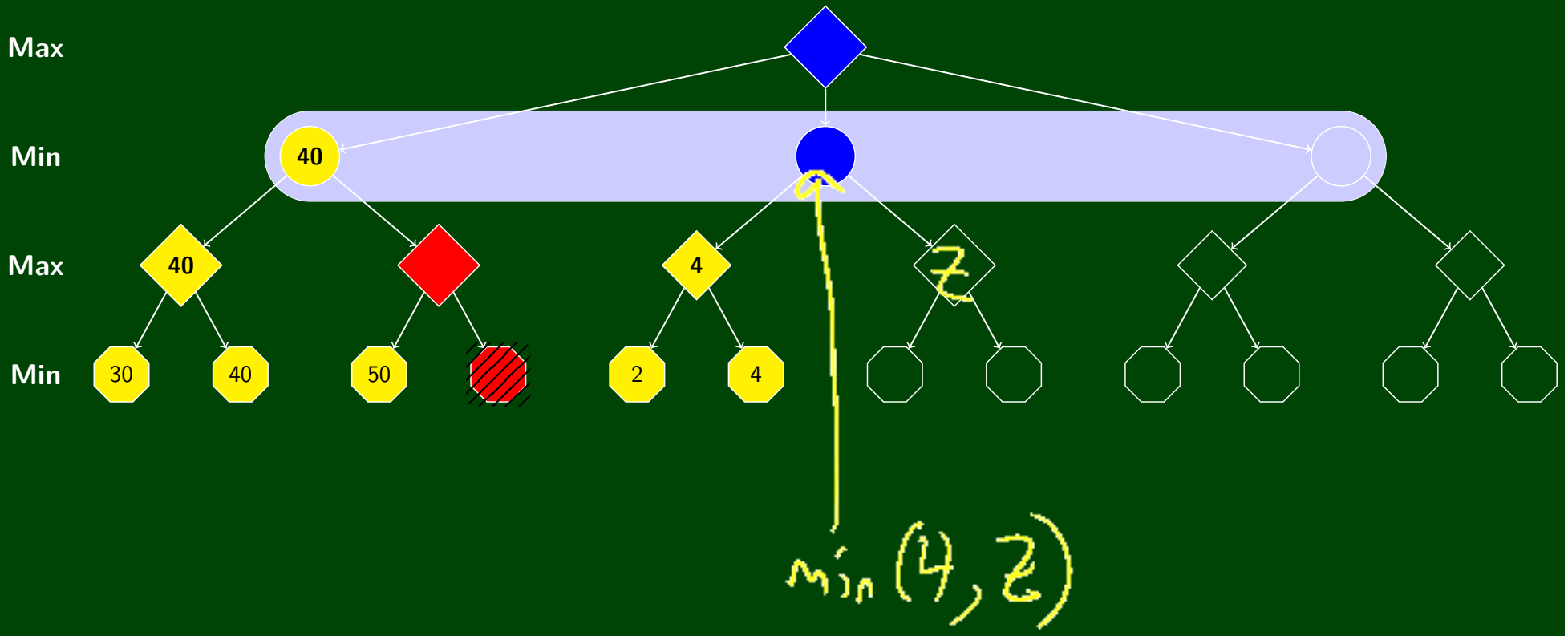


Do we check the next node?

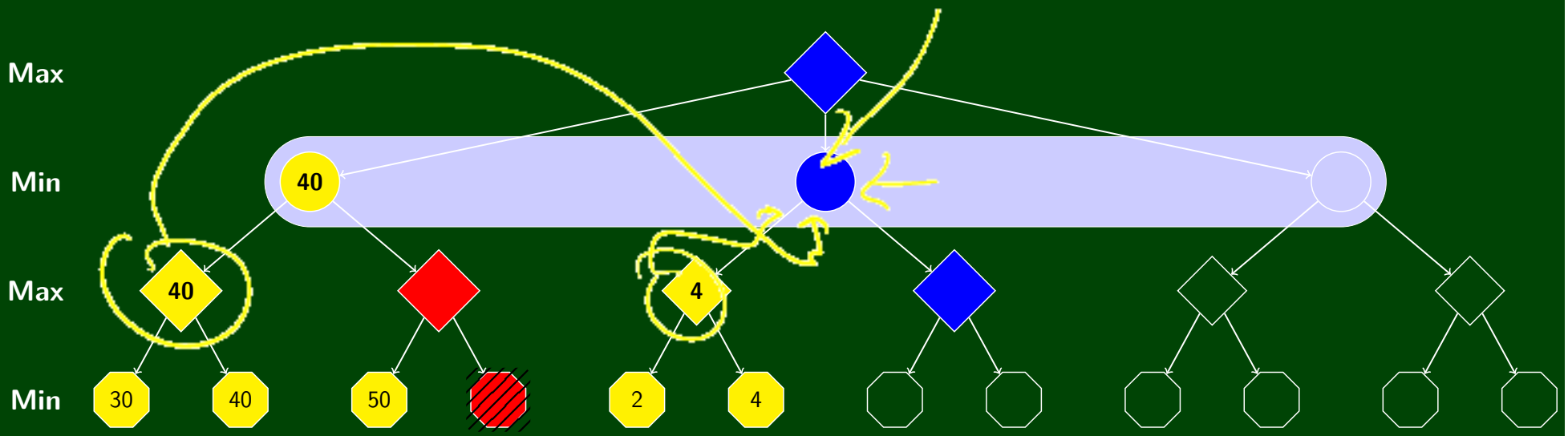
Max will choose $x \geq 50$ which is already worse than the 40.

The current bounds are $[50, 40]$. Don't bother.

Pruning



Pruning



Do we check the next node?

Min will choose $x \leq 4$ which is already worse than the 40.

The current bounds are $[40, 4]$. Don't bother.