

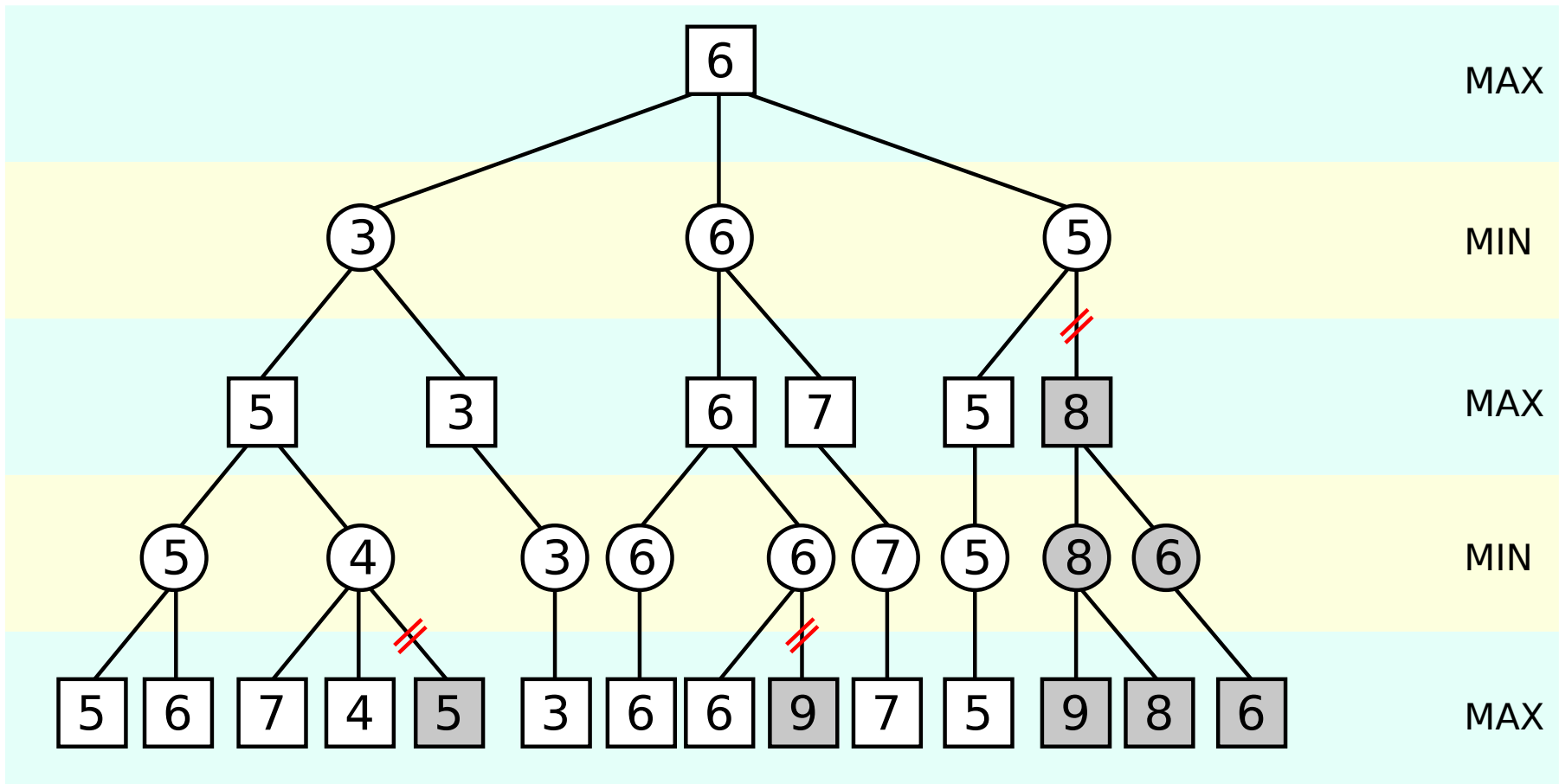
# chess

- sounds boring and trivial
- but it really isn't

# what i did the in the last couple of weeks

- pseudo-legal move generator (filter out illegal moves later), verified by a perft() test suite
- basic alpha-beta search (with buggy TT support)
- some wonky parallelization that's not really worth talking about
- some tooling + bare minimum uci implementation for testing purposes only

# minimax/alpha-beta



# vectorization

- there **is** potential for vectorization in the move generator (not so much in the search itself)
- but you absolutely have to use bitboards
- might require a major rewrite of all move generating code even if you are already using bitboards

# parallelization

- is **not** trivial since alpha-beta is inherently sequential
- (almost) all serious approaches are SMP and require a shared transposition table
- you will most likely have to do your own thread management
- naive approaches like distributing root moves won't stand a chance vs. a decent sequential alpha-beta

# lessons learned

- start early :)
- utilize test suites – chances are you forgot to test several rare edge cases anyways
- do the same for debugging search(), testing whole games is way too inefficient
- i.e. use best move test suites, mating problems etc.