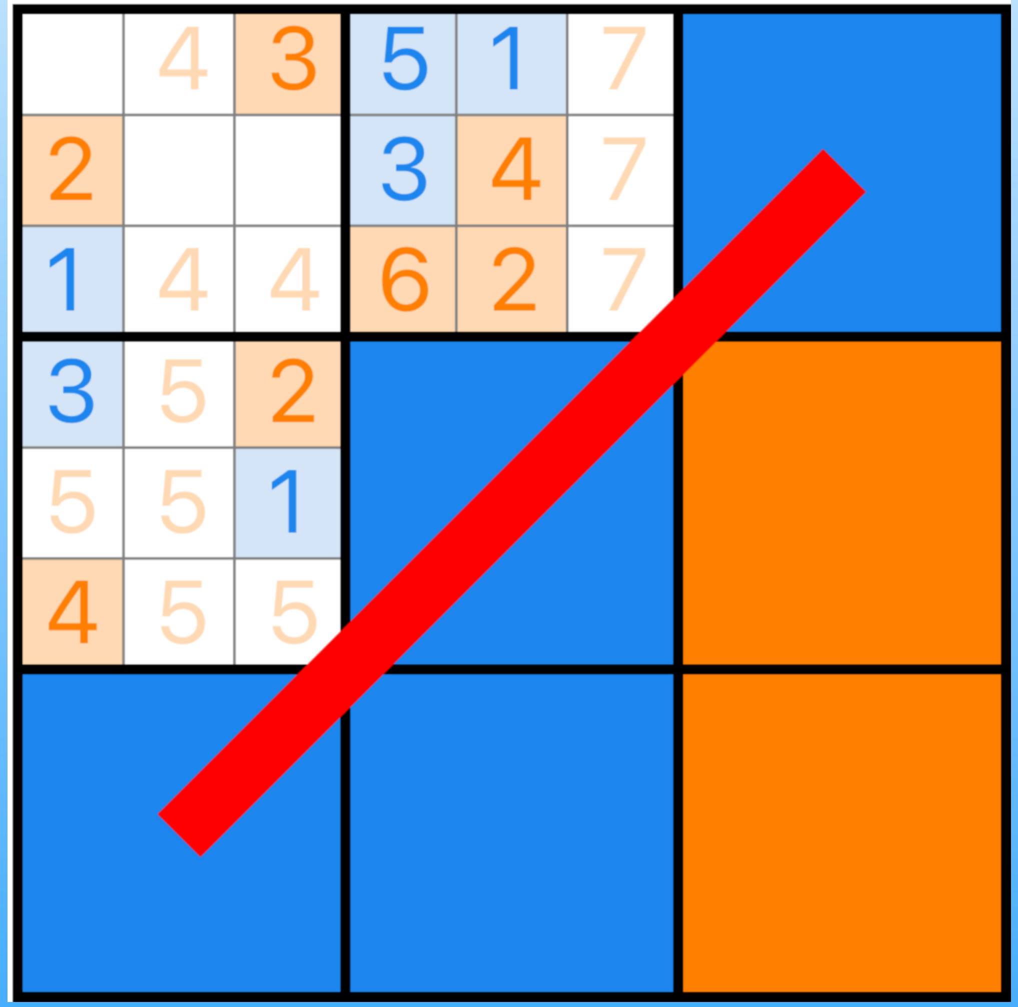
Monte Carlo Tree Search

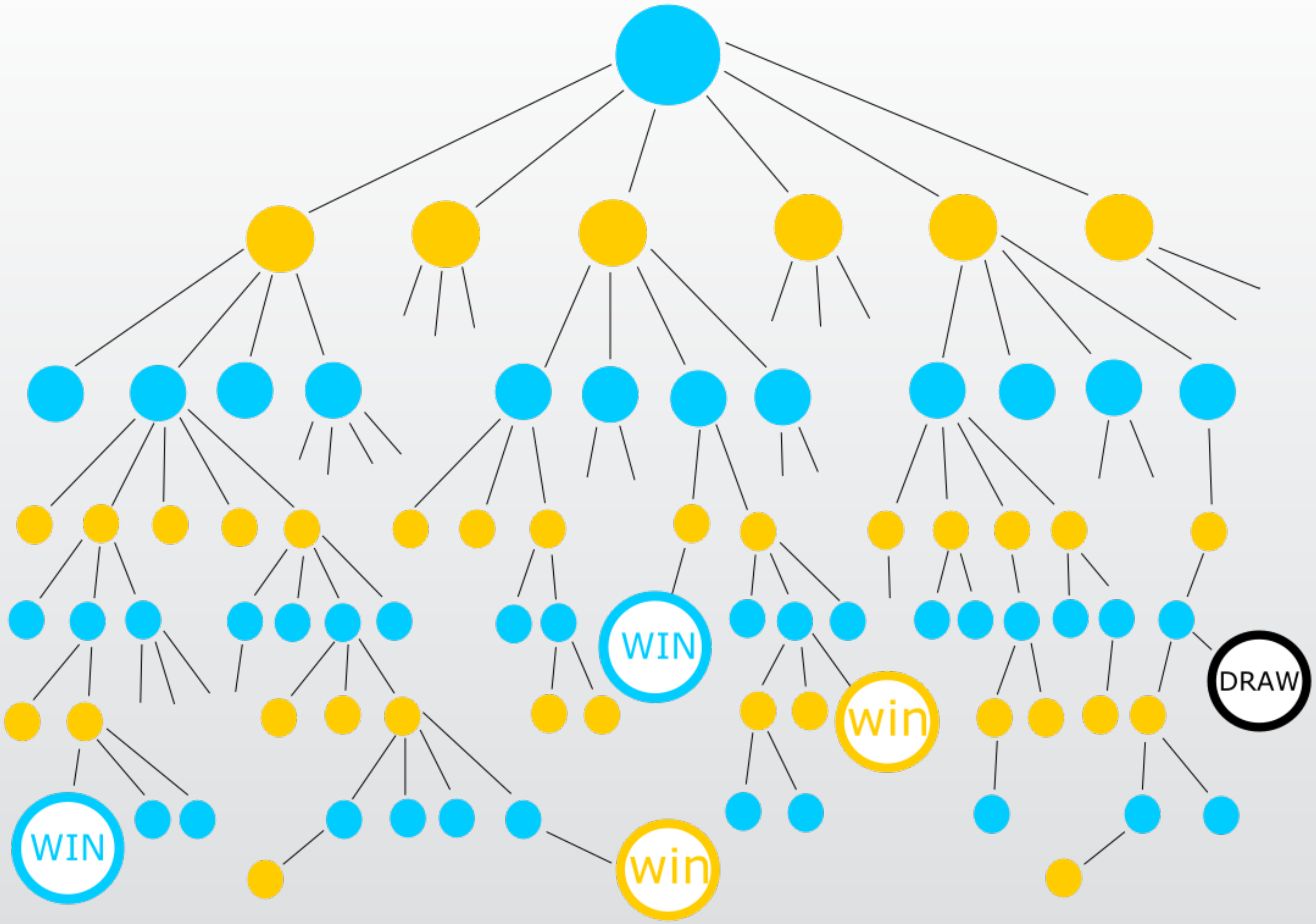
By Max Khrapov

Sansumoku



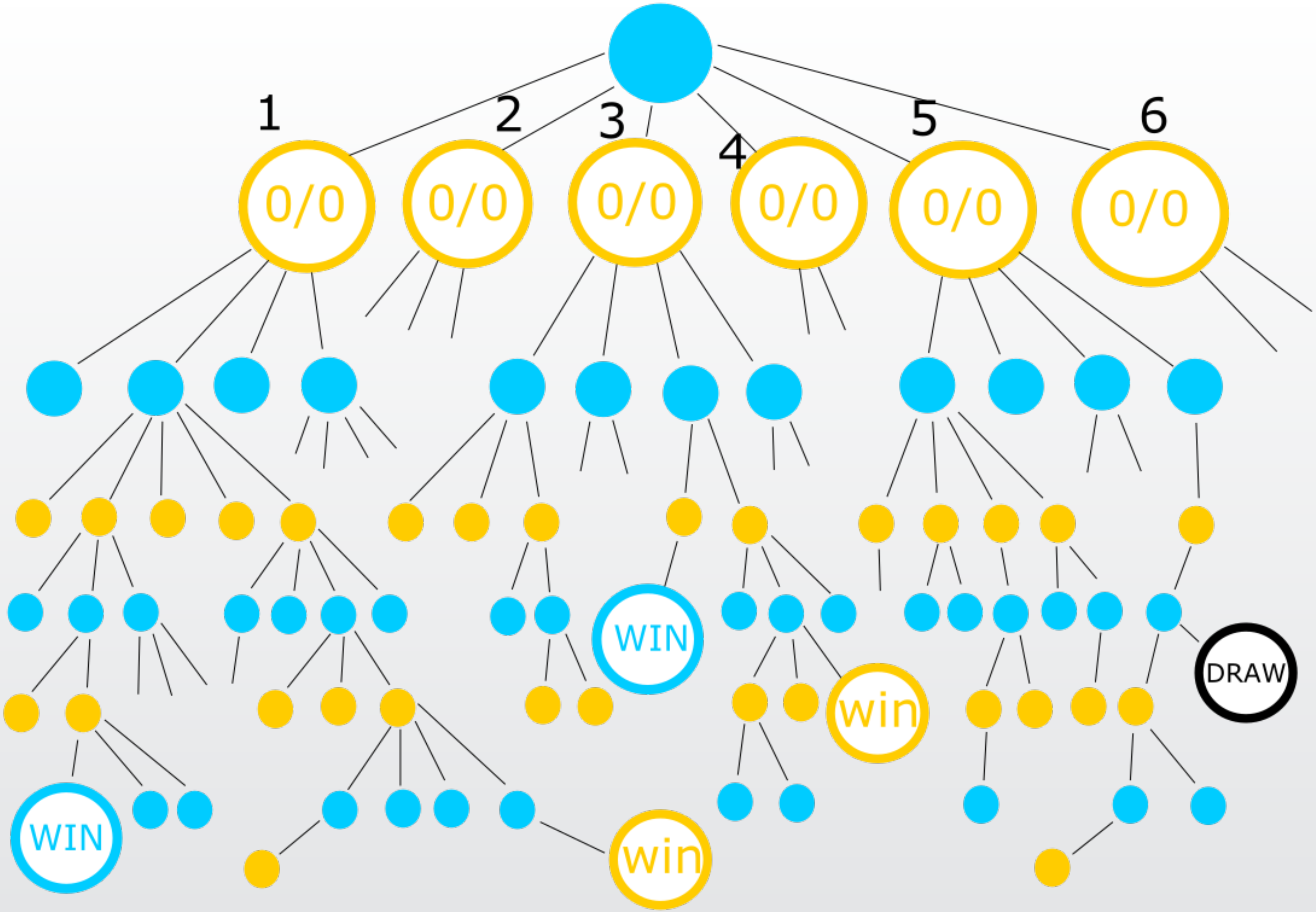
- Two Player Strategy Board Game
- iOS app (working on macOS version)
- https://www.sansumoku.com
- Open Source
- https://github.com/mkhrapov/sansumoku

Game tree search



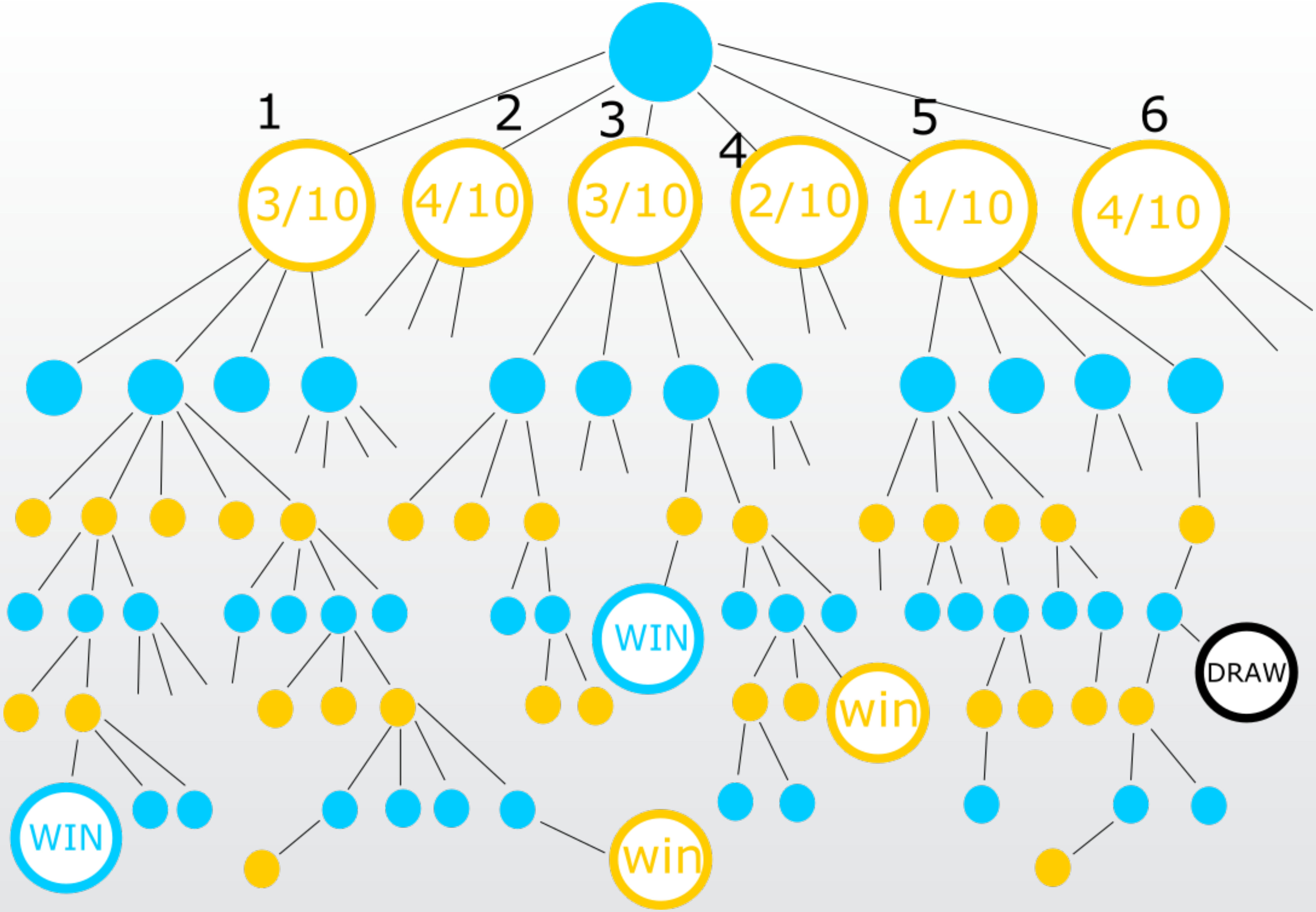
Minimax

- Requires either a brute force search
- Or a position evaluation function
- Neither is applicable to Sansumoku



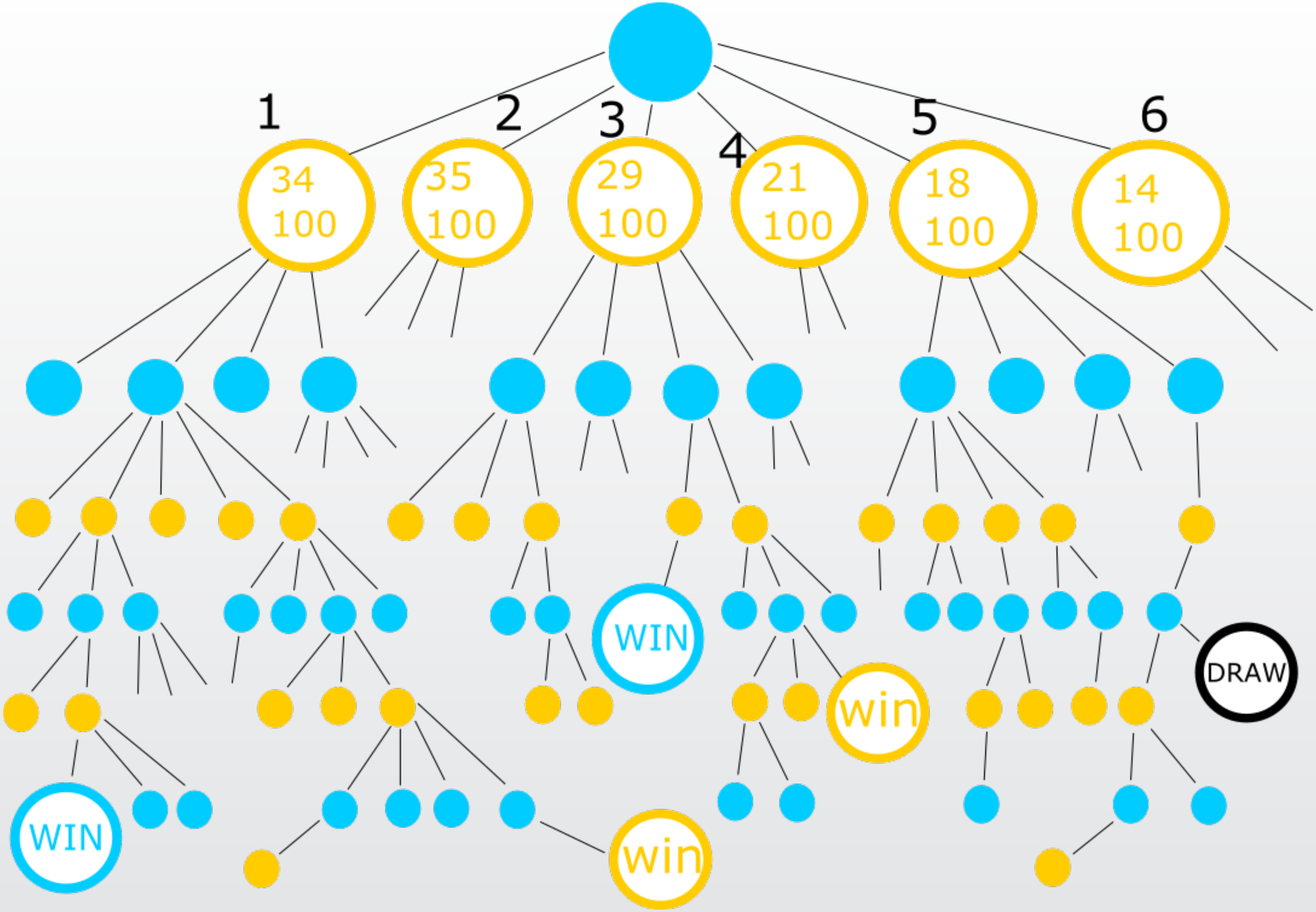
Monte Carlo Tree Search

Statistical evaluation of the game moves



Monte Carlo Tree Search

Statistical evaluation of the game moves



Monte Carlo Tree Search

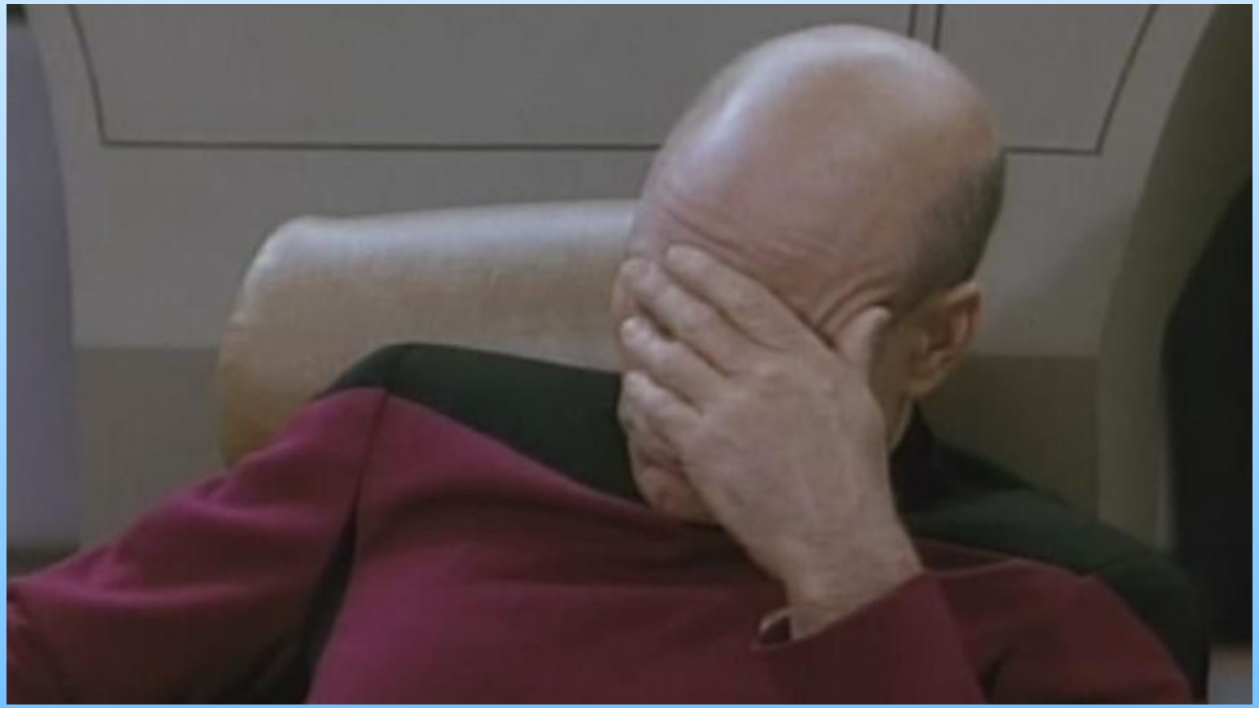
Statistical evaluation of the game moves

How long should I search?

Players	Win	Lose	Draw
1000 iterations vs 100 iterations	65	31	4
2000 iterations vs 1000	55	43	2

Apple GameplayKit on iOS

- GameplayKit https://developer.apple.com/documentation/gameplaykit
- GKMinimaxStrategist
- GKMonteCarloStrategist



Is my implementation better than Apple’s?

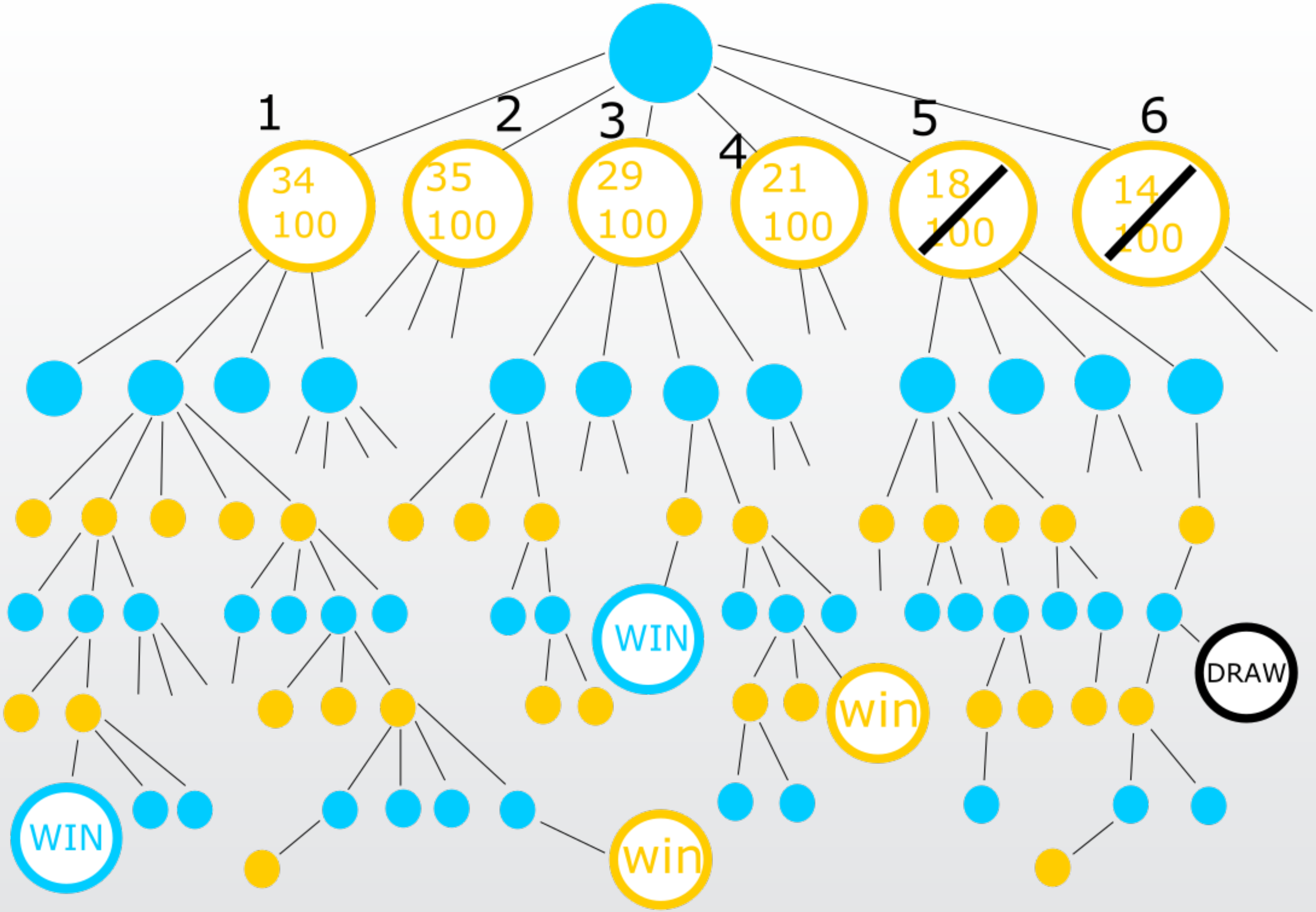
Apple vs	Win	Lose	Draw
Random	95	5	0
Basic	83	11	6
MCTS			

Is my implementation better than Apple’s?

Apple vs	Win	Lose	Draw
Random	95	5	0
Basic	83	11	6
MCTS	6	94	0

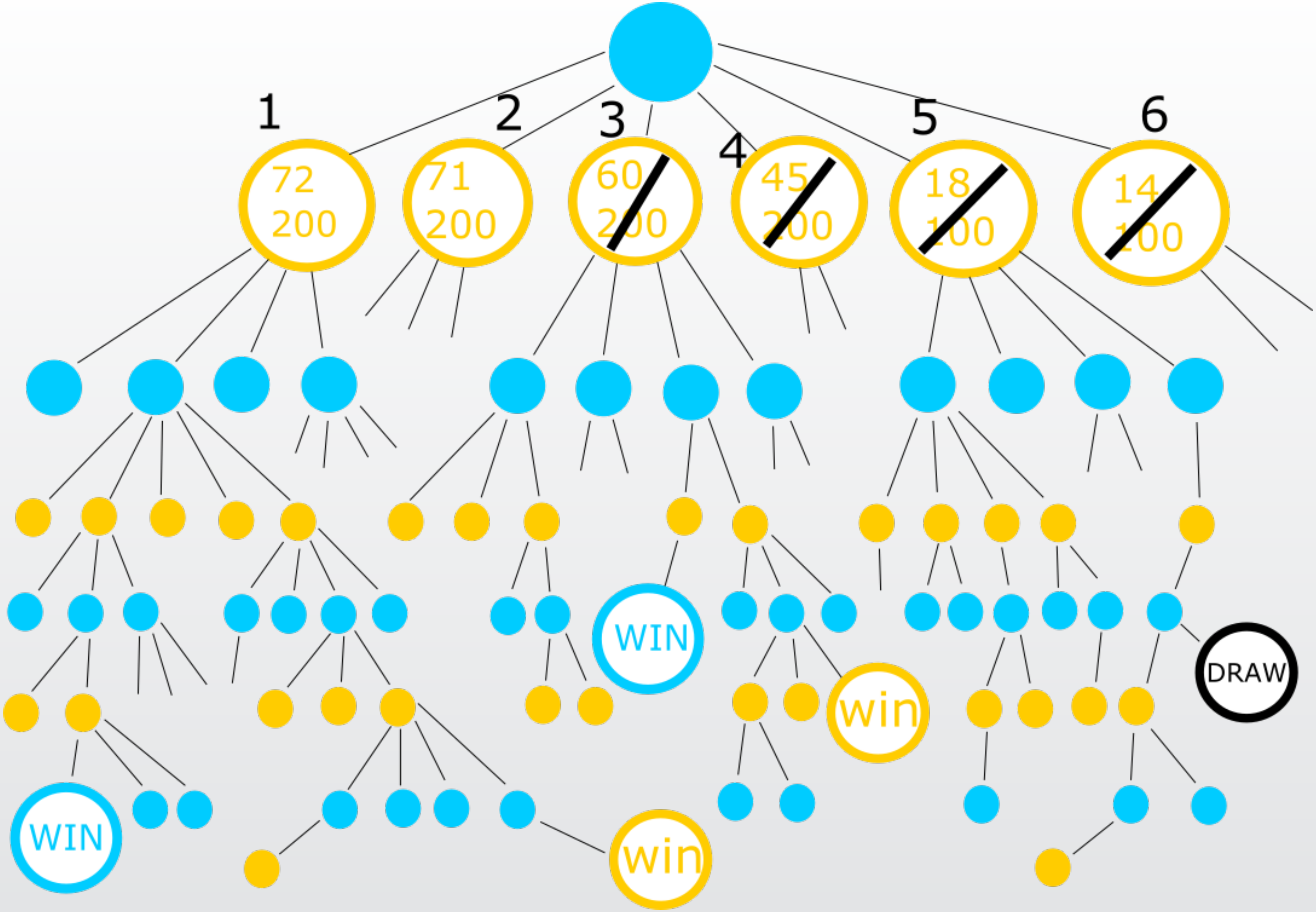
Future Improvements

- Multithreaded Search
- Switch to minimax
- Make play out more intelligent
- Spend less time on bad moves



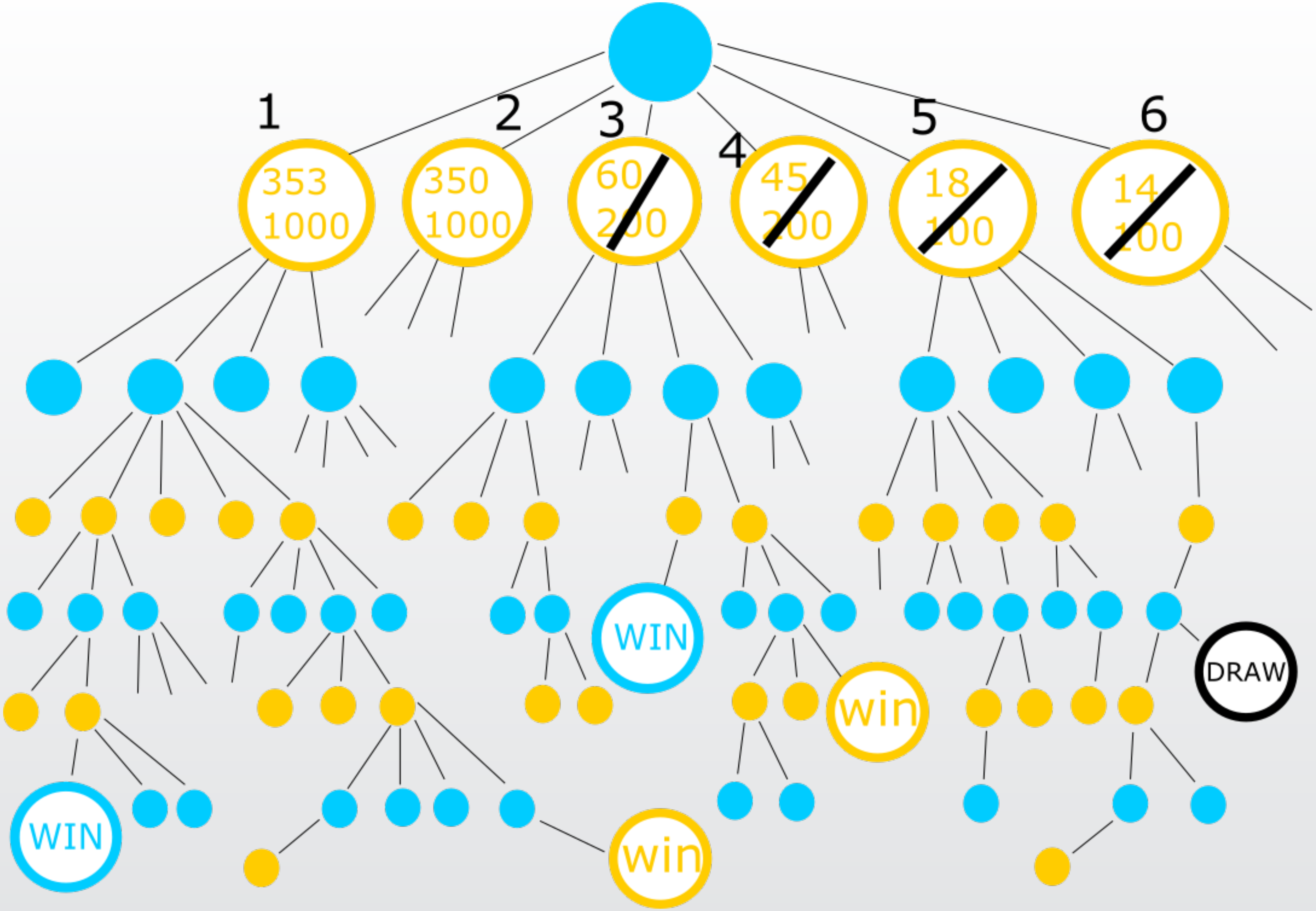
Monte Carlo Tree Search

Statistical evaluation of the game moves



Monte Carlo Tree Search

Statistical evaluation of the game moves



Monte Carlo Tree Search

Statistical evaluation of the game moves

Sansumoku