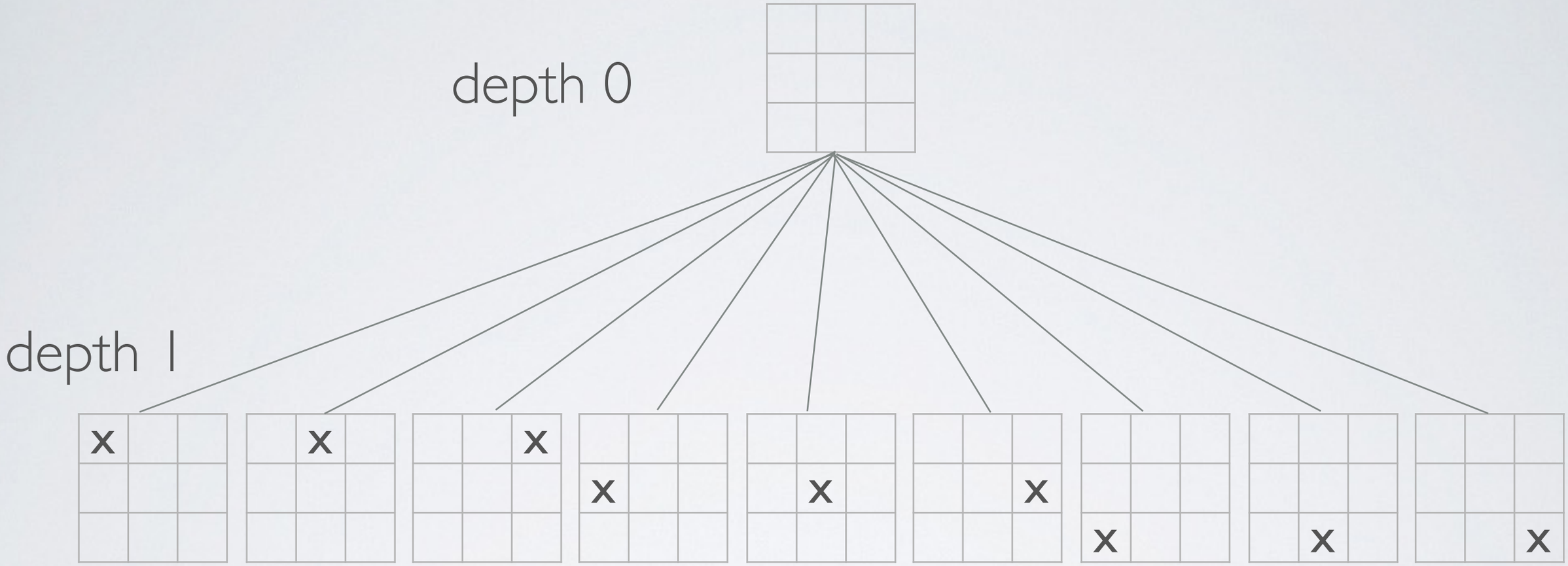
PROJECT II

E. Gerlitz, A. Popkes, P. Wenker, K. Patel, N. Lutz

STRUCTURE

- Tic tac toe game tree
- Minmax algorithm
- Minmax for Connect4
- Breakout
- Path Planning
  - Dijkstra's algorithm
  - A* algorithm

THE TIC TAC TOE GAME TREE



$$\underline{\text{Upper bound}}$$

$$\text{Total number of nodes} = 1 + 9 + 9 \cdot 8 + 9 \cdot 8 \cdot 7 + \ldots$$

$$= \sum_{i=0}^{9} \frac{9!}{(9-i)!} = 986410$$

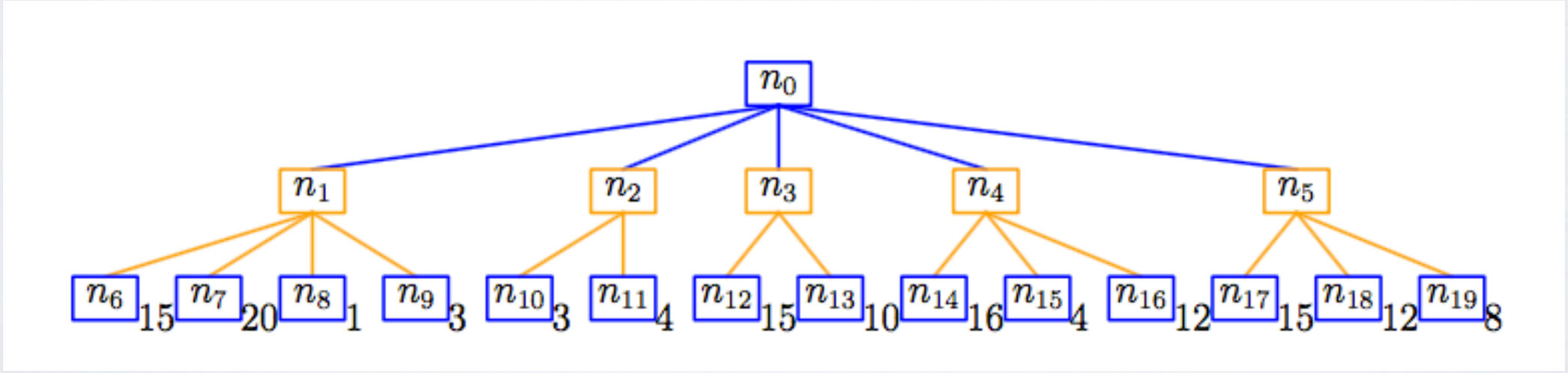
THE TIC TAC TOE GAME TREE

- number nodes in the tree: 549946
- number of times X wins: 131184
- number of times O wins: 77904
- average branching factor: 1.86

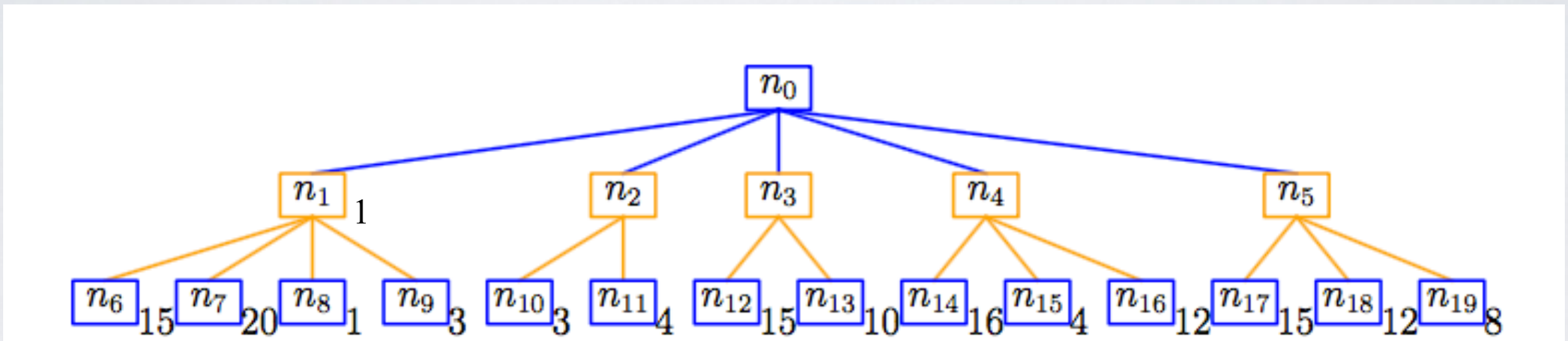
MIN MAX ALGORITHM

recursive computation

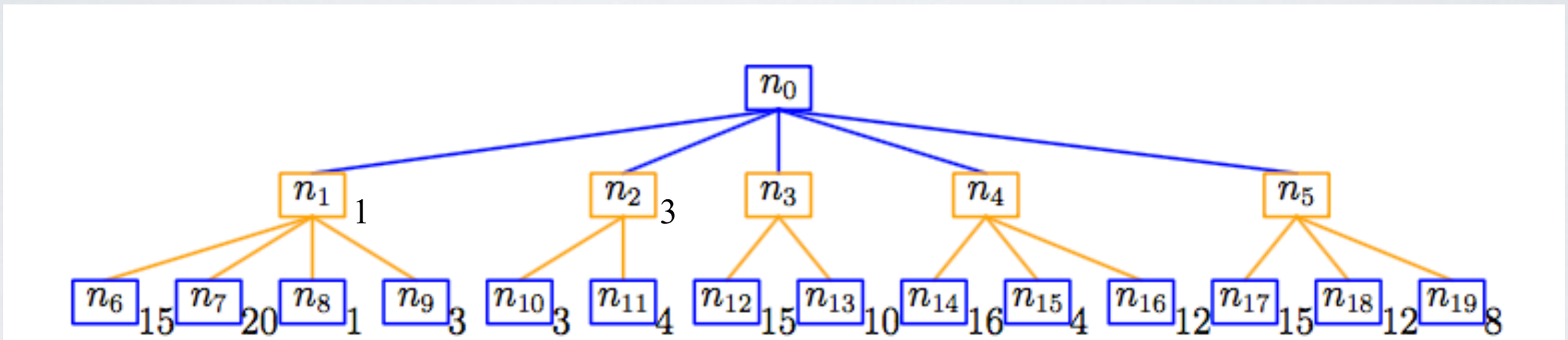
$$mmv(n) = \begin{cases} u(n) & \text{if } n \text{ is a terminal node} \\ \max\limits_{s \in Succ(n)} mmv(s) & \text{if } n \text{ is a } MAX \text{ node} \\ \min\limits_{s \in Succ(n)} mmv(s) & \text{if } n \text{ is a } MIN \text{ node} \end{cases}$$



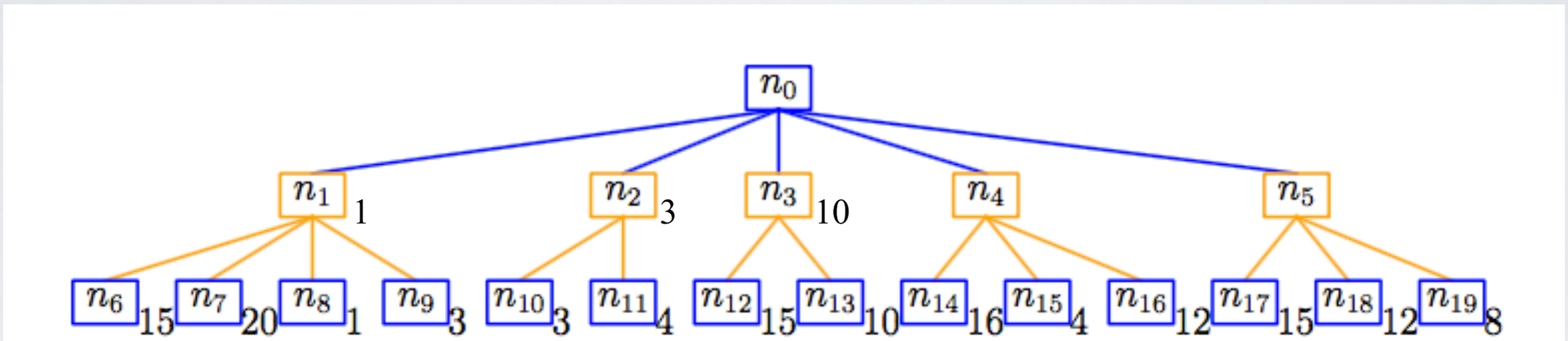
MIN MAX ALGORITHM



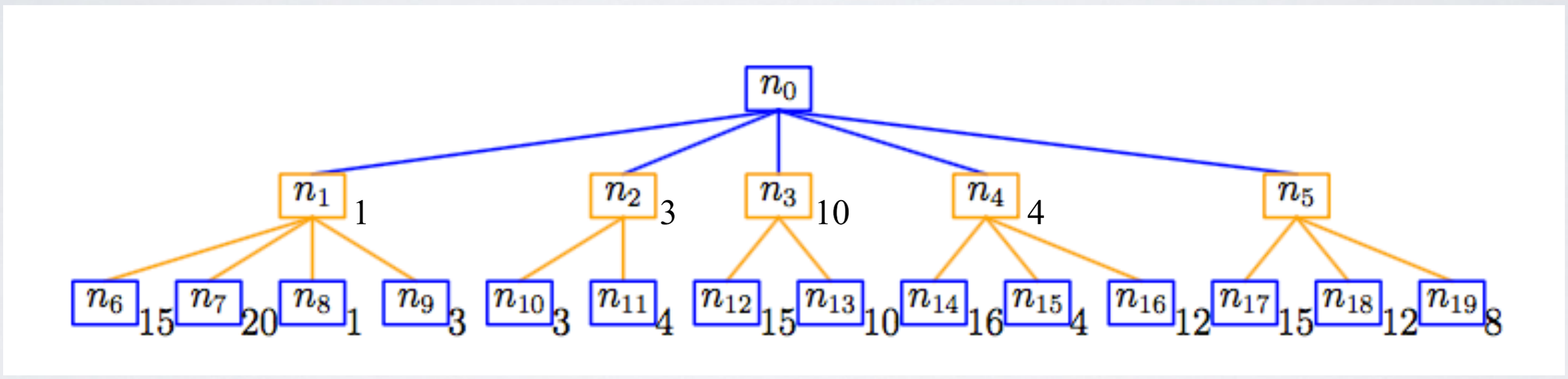
MIN MAX ALGORITHM



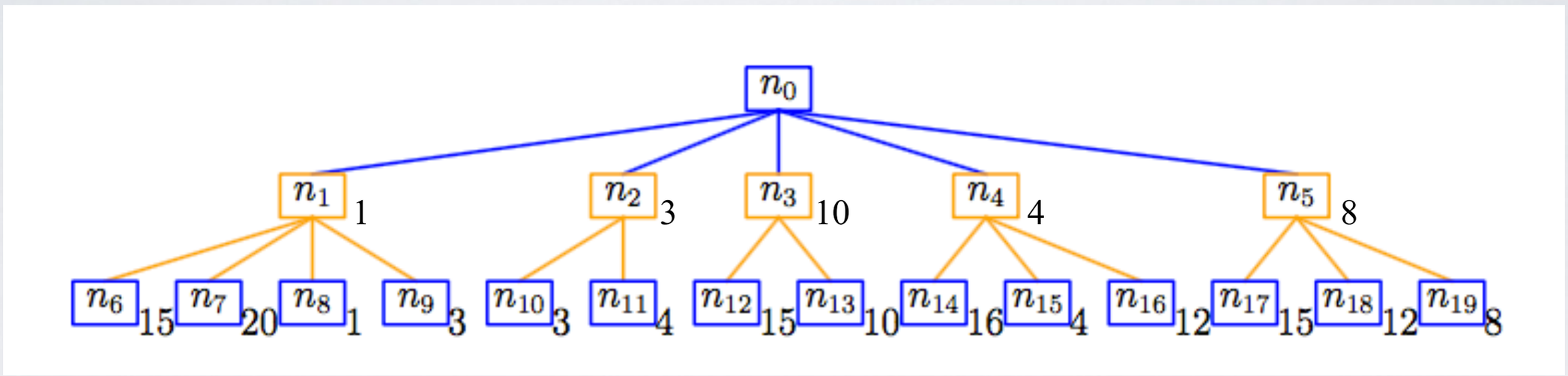
MIN MAX ALGORITHM



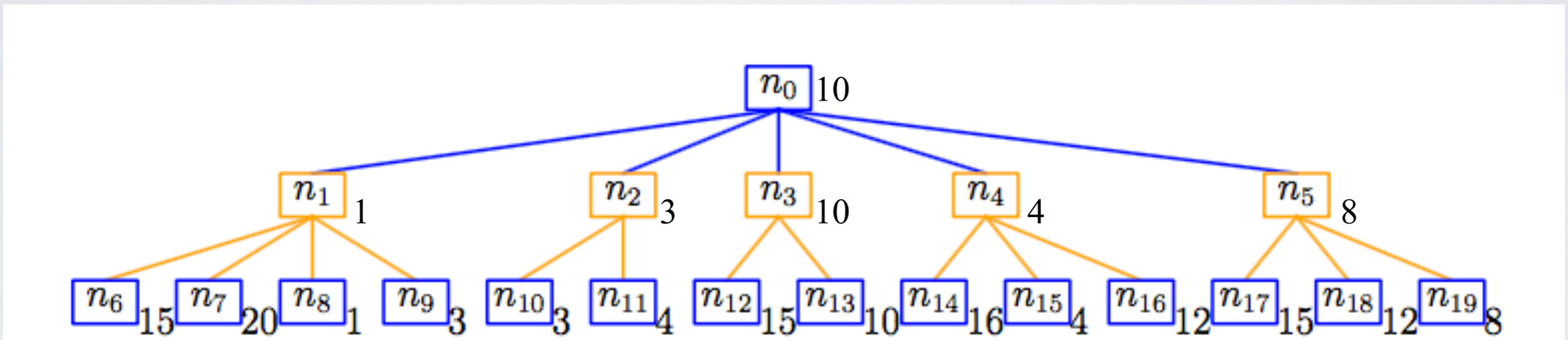
MIN MAX ALGORITHM



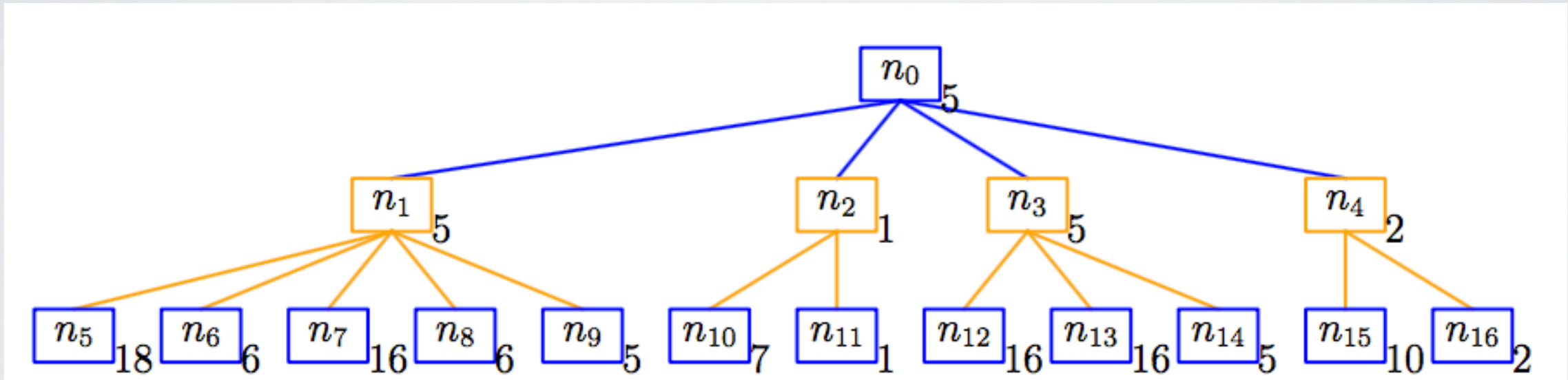
MIN MAX ALGORITHM



MIN MAX ALGORITHM



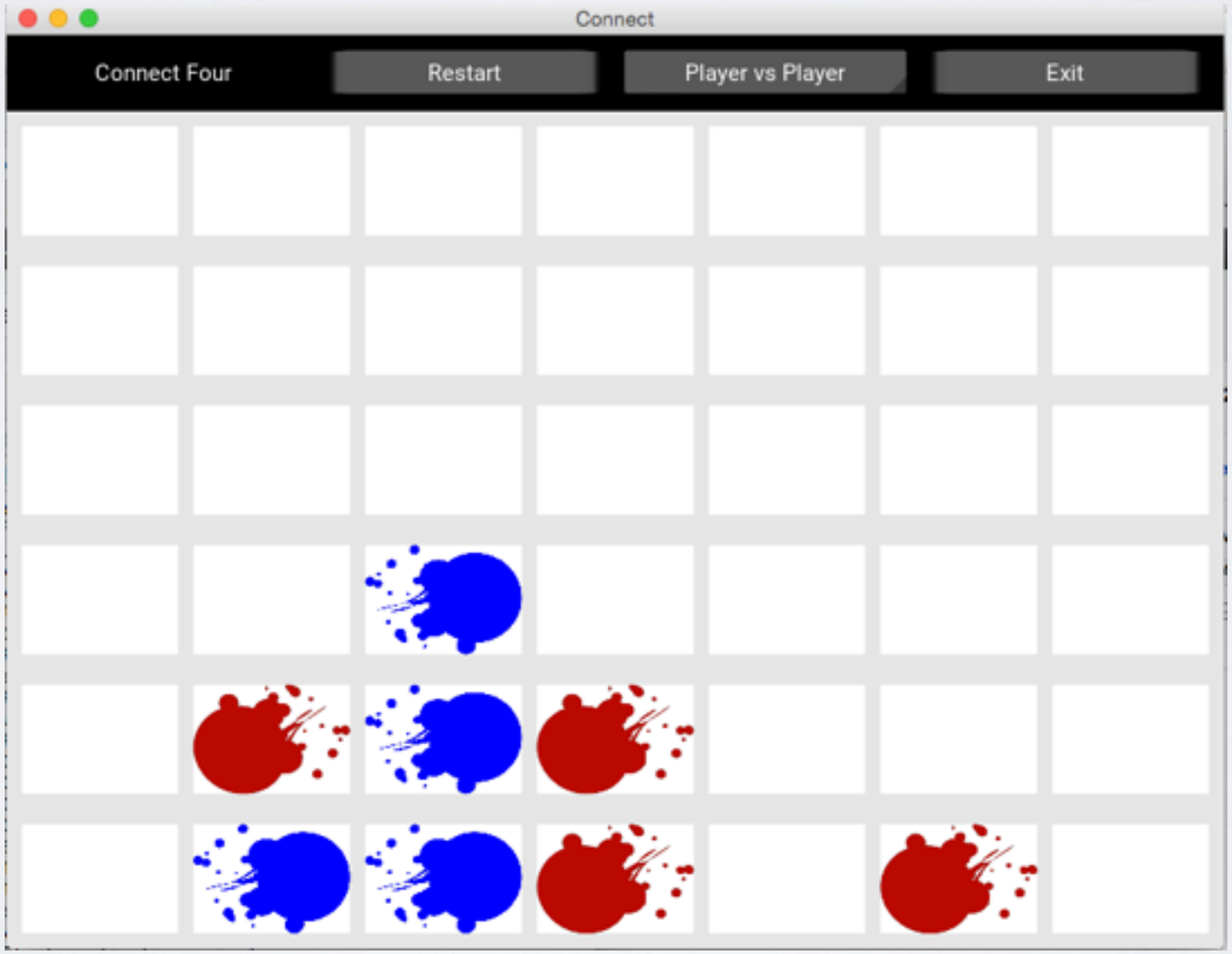
MIN MAX ALGORITHM



- Idea: Store max & average of next level
- Problem: If we assume that the opponent always chooses the minimum, there is no point in storing another value

MIN MAX FOR CONNECT 4 - HEURISTIC

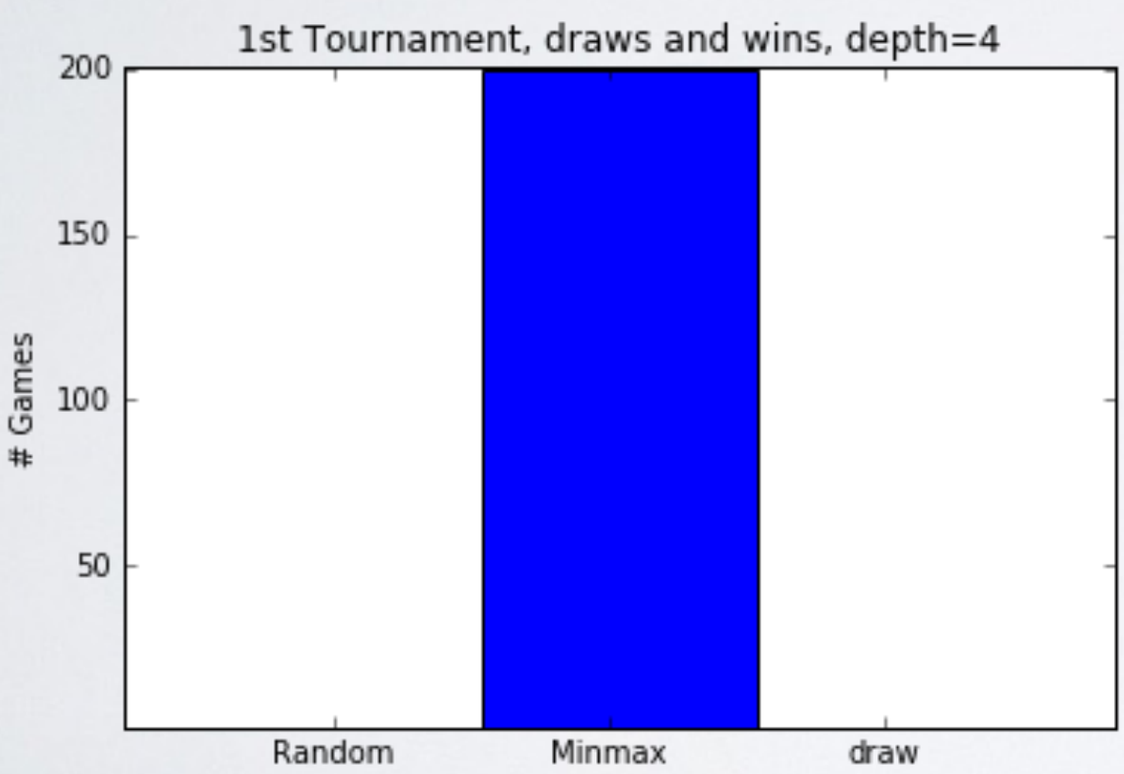
Score = weighted occurrences of 4, 3, 2 in a row of player
- weighted occurrences of 4, 3, 2 in a row of opponent

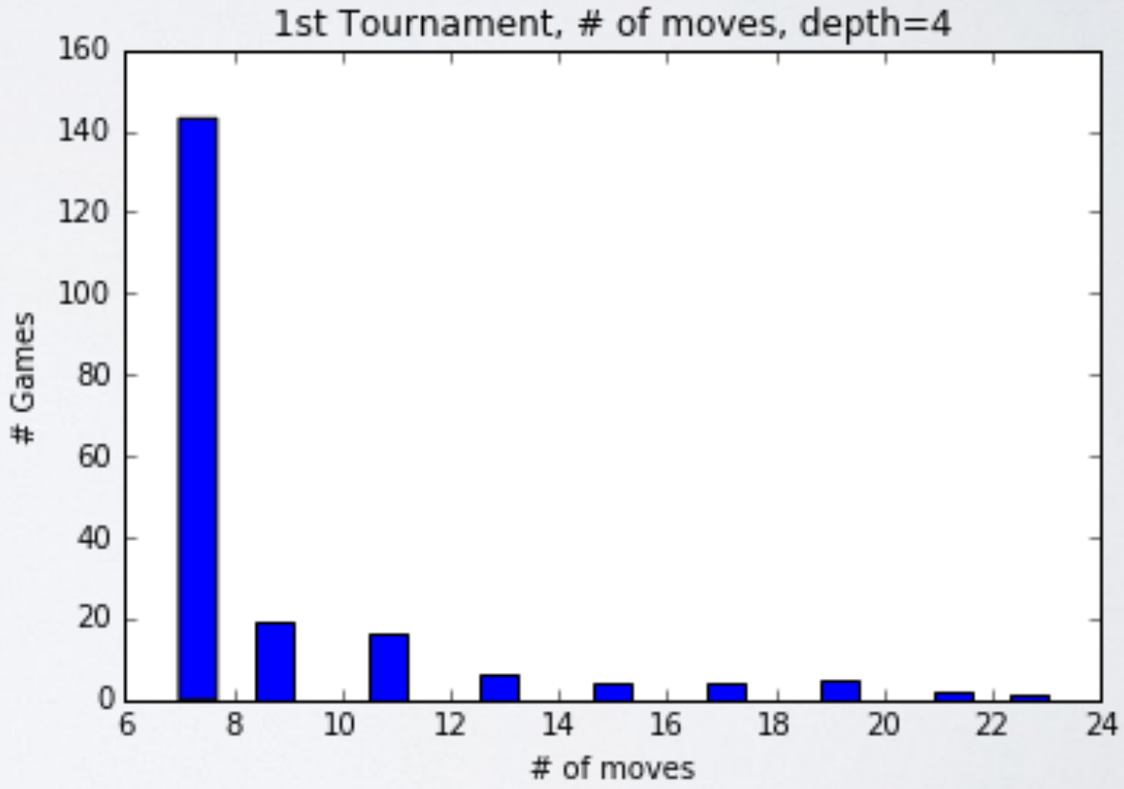


MIN MAX FOR CONNECT 4

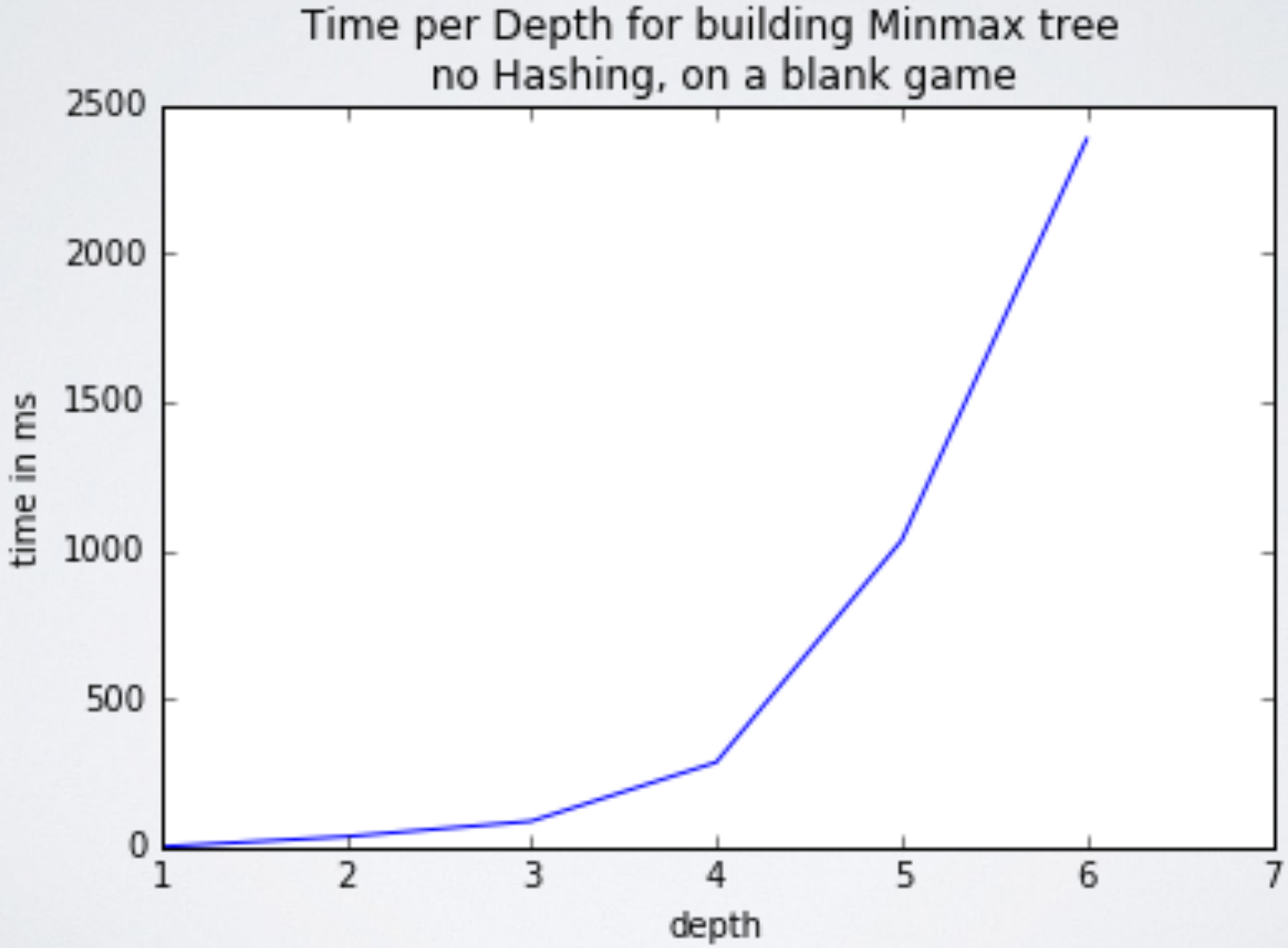
Depth restricted Minmax using:

- alpha-beta pruning
- score hashing

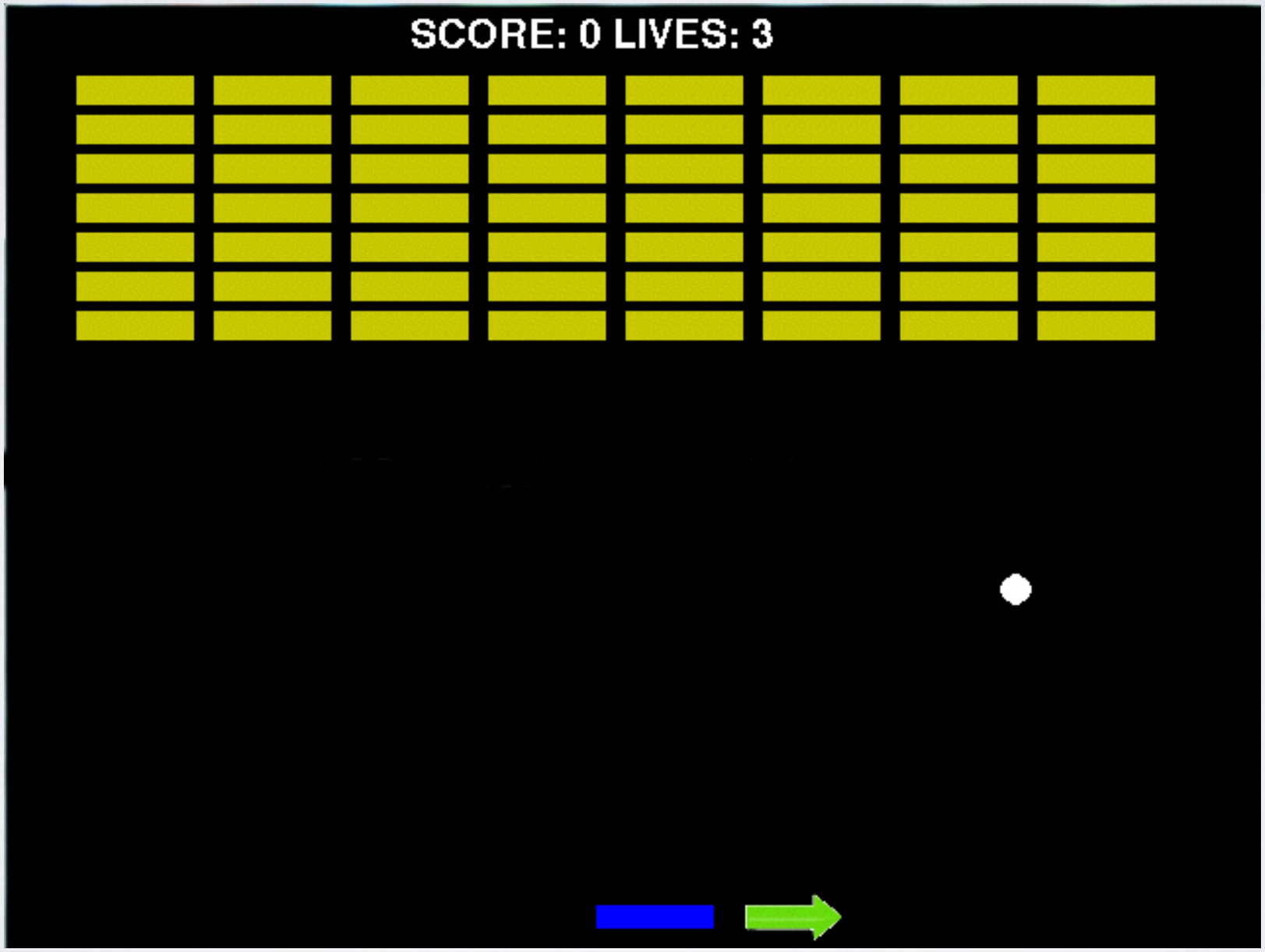




MIN MAX FOR CONNECT 4 - TIME COMPARISON



BREAKOUT - SIMPLE BALL FOLLOWER

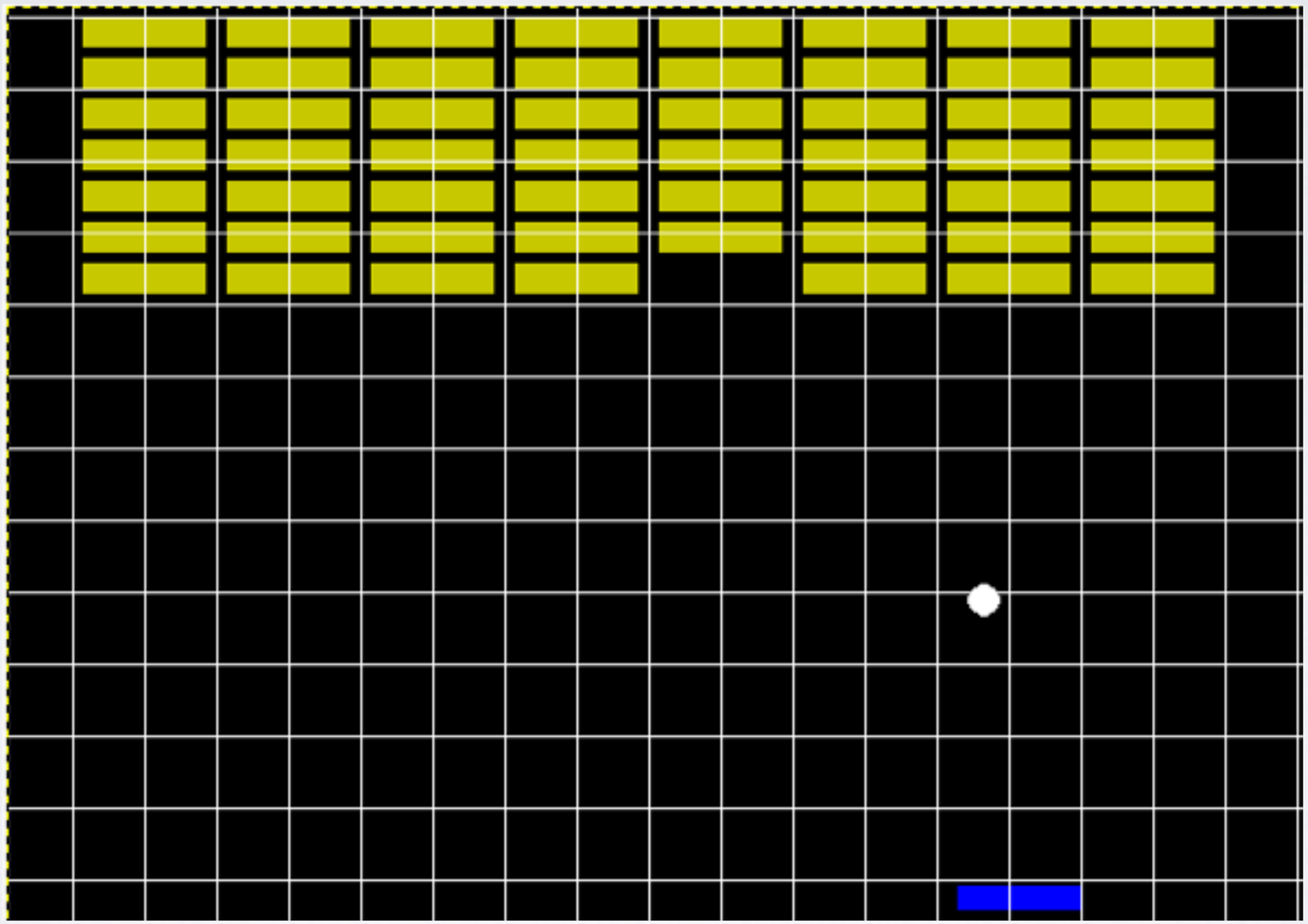


BREAKOUT - REINFORCEMENT LEARNING

- 3 Main Components:
  - Agent: Learner/Decision Maker
  - Environment: Everything the agent interacts with
  - Actions: What the agent can do

- Need to learn Action-Value function: $Q^{\pi}(s, a)$

REINFORCEMENT LEARNING - STATE & REWARD FUNCTION



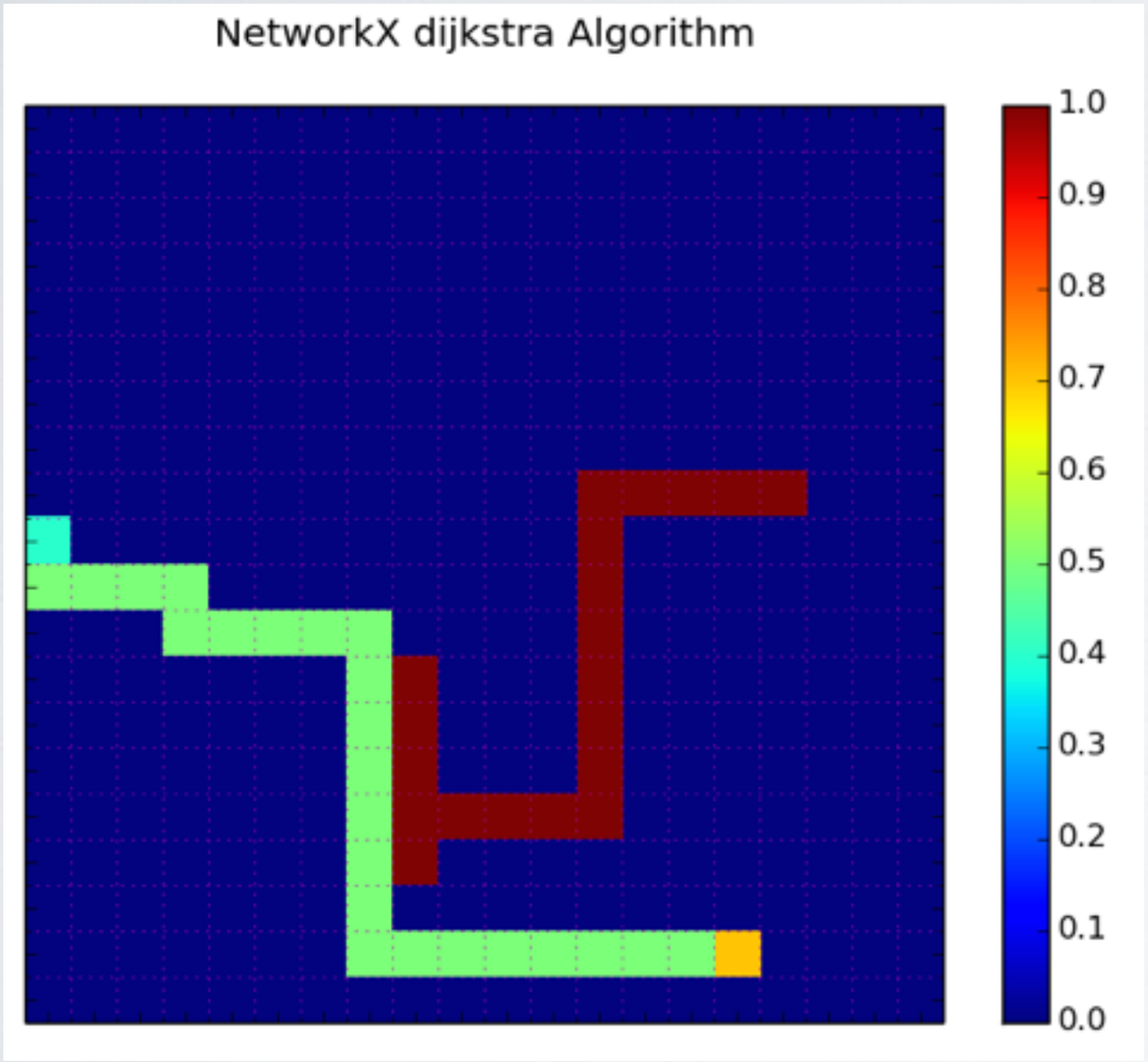
$$\text{reward} = \begin{cases} -10000 & \text{if episode ends} \\ -\text{distance to ball} + \text{score} & \text{else} \end{cases}$$

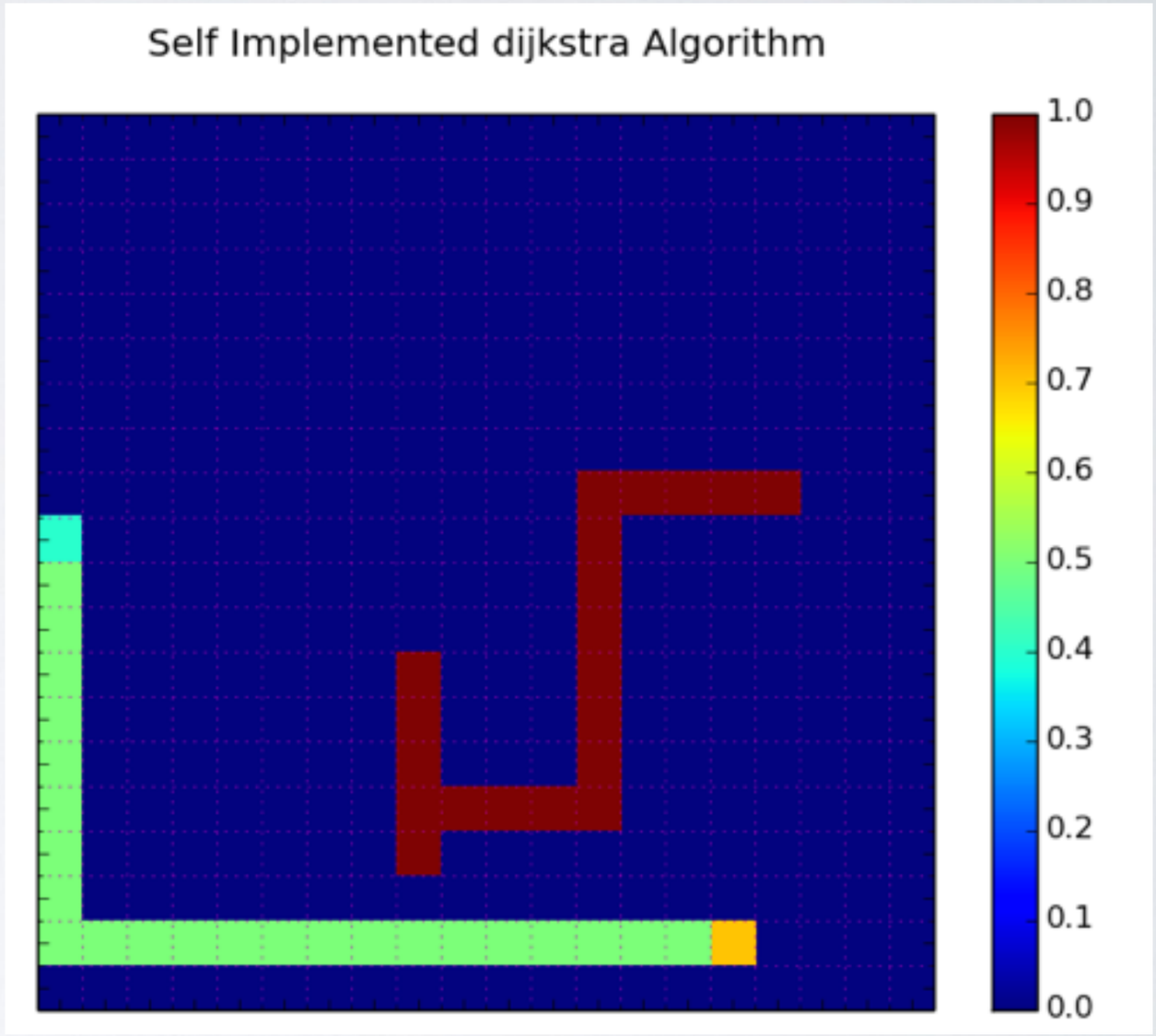
BREAKOUT - INCREASING BALL SPEED

- Both AI controllers cope with the increase in speed of ball until the point where the ball speed exceeds the paddle speed

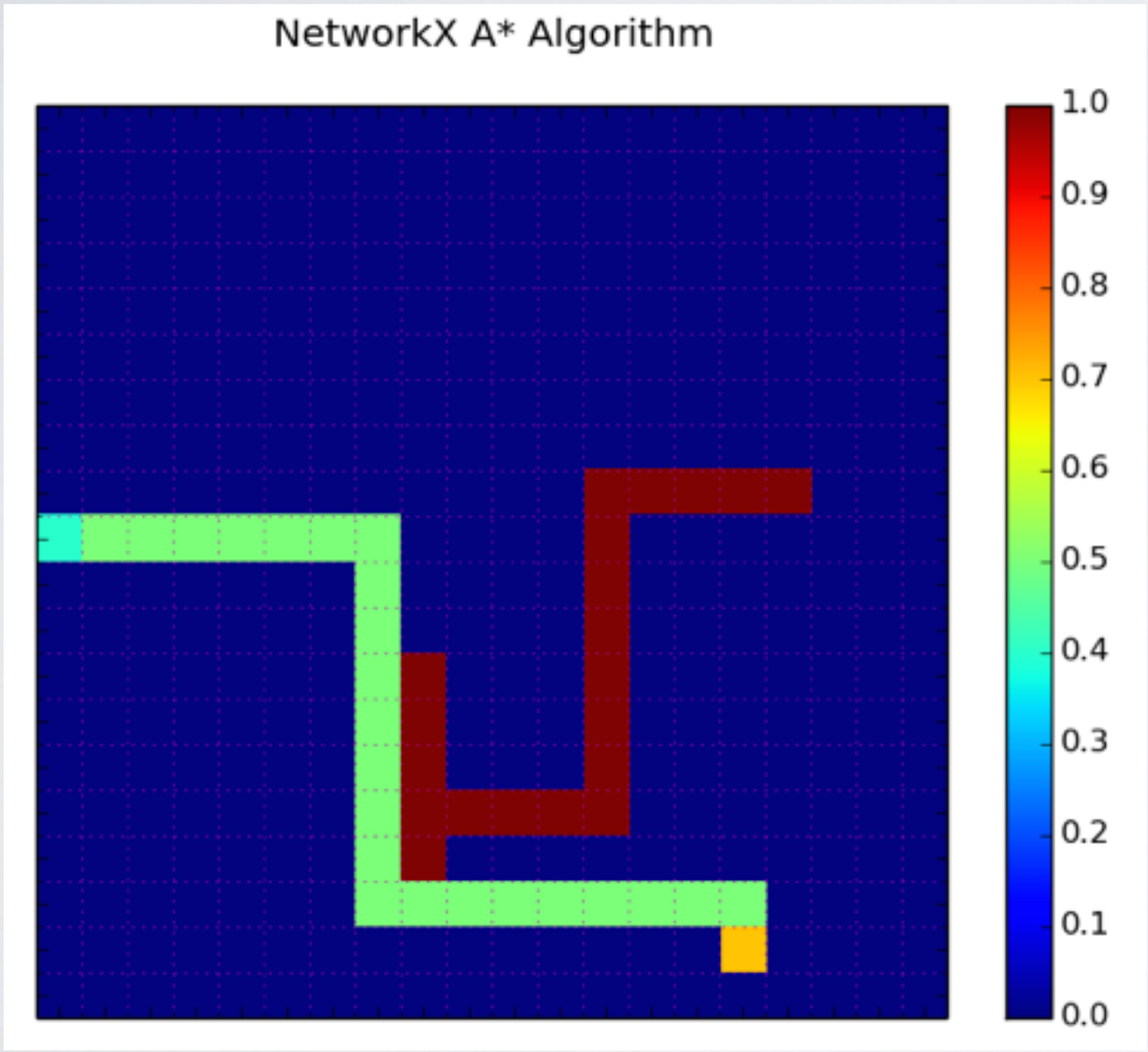
- If you increase the speed of paddle, the episodes last longer

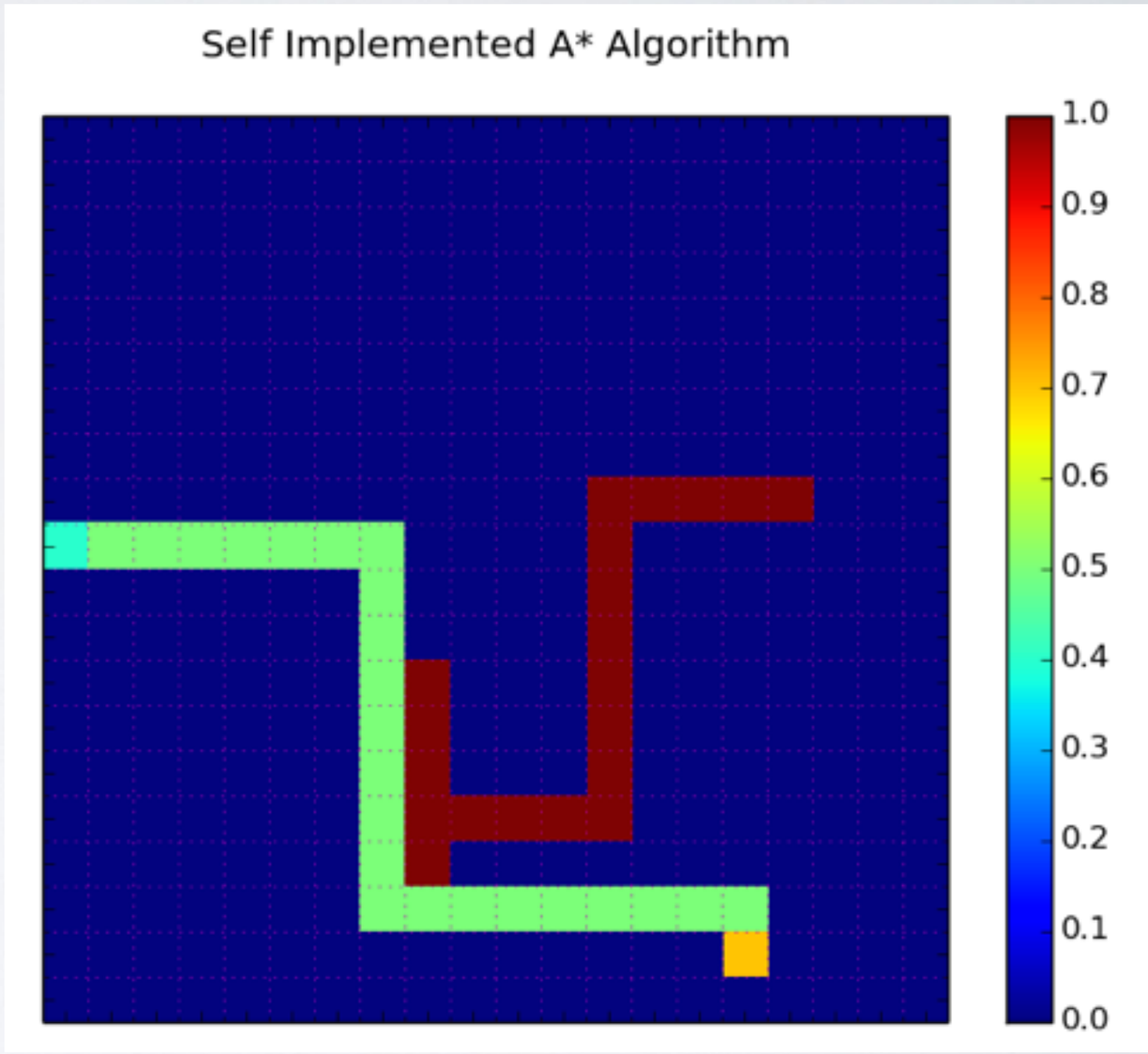
PATH PLANNING - DIJKSTRA'S ALGORITHM





PATH PLANNING - A* ALGORITHM





PATH PLANNING - A* ALGORITHM