SUMMARY:

Research paper: **Game Tree Searching by Min / Max Approximation** by Ron Rivest, MIT (Fun fact, Ron Rivest is the R is in the RSA cryptographic protocol).

Paper techniques: The paper tries to solve the combinatorial problem associated with expansion of nodes in a Minmax algorithm. Traditionally, Min and Max operators are used to drill down the tree structure. However, an approximating generalized mean-value operator or a function, such as a p-mean, can be used as proxy in such evaluations. This helps in performing a sensitivity analysis and expanding the node that is expected to have the largest effect on the value at the root node, and in performing a partial differential w.r.t to each potential node for expansion using chain rule.

Subsequent sections define and explore different ways of Game tree searching. Within a set of finite time and resources, it is argued that for large game trees, heuristics approximations are needed. This paper deals with iterative heuristics, wherein, at each step a tip node (or leaf) of the current tree is chosen, and the successors of that tip node are added to the tree. Then the values provided by the static evaluator at the new leaves are used to provide new backed-up values to the leaves' ancestors. The tree grown by an iterative heuristic need not be of uniform depth: some branches may be searched to a much greater depth than other branches.

In further sections, penalty-based iterative search methods are evaluated. In this approach, we assign a nonnegative "penalty" (or "weight") to every edge in the game tree such that edges representing bad moves arc penalized more than edges representing good moves. Different examples are demonstrated to make a case in point. Finally, the implementation of such iterative heuristic is suggested along with the pseudo code.

Paper Results:
The game of Connect-Four was used for experimentation. Iterative heuristic can produce play superior to that produced by minimax search with alpha-beta pruning, for the same number of calls to the underlying "move" operator. However, when the CPU time is used as a limiting resource, then minimax search with alpha beta pruning seems to play better.

Resource bound per turn	MM wins	AB wins	Ties
1 second	41	46	11
2 second	40	42	16
3 seconds	36	44	18
4 seconds	39	52	7
5 seconds	30	55	13
Total	186	239	65
1000 moves	47	35	16
2000 moves	50	35	13
3000 moves	42	47	9
4000 moves	49	42	7
5000 moves	61	31	6
Total	249	190	51