CSE 332: Data Structures and Parallelism

Section 8: P3 and Search

0. MiniMax

(a) Will the move returned by an Alpha-Beta search (always), or (never) be better than the move returned by a MiniMax search on the same board with the same search parameters?

(b) Explain why we *negate* the result of the recursive call in MiniMax.

(c) Why is MiniMax "naturally parallelizable", while Alpha-Beta is not?

1. Cutoffs

Provide a short diagram or description to explain the following parameters from P3:

(a) `ply`

(b) `cutoff`

(c) `divideCutoff`

(d) `PERCENTAGE_SEQUENTIAL`

2. Efficiency

Circle the **most efficient** option from each pair of possible implementation strategies for P3:

(a) To create threads for each move in a `List<M>` during Parallel Minimax:

Create threads in a `for` loop **OR** Create threads with divide-and-conquer

(b) To pass copies of boards to these threads:

Copy the board *inside* the thread **OR** Copy the board *before* passing it to the thread

(c) To evaluate a list of moves using Alpha-Beta pruning:

Evaluate the moves in the order provided **OR** Sort the moves best-first, then evaluate in sorted order

3. Alpha-Beta

Determine the value of the root node after running Alpha-Beta on the following tree (and cross out pruned branches/nodes):