

CSE 332: Data Structures and Parallelism

Section 6: P3 and Search Solutions

0. MiniMax

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

Solution:

We do this in order to simulate the idea of switching between your perspective and your opponent's perspective. By using the assumption that your opponent is playing optimally, if you negate the result of the recursive call, then on your opponent's turn, they will choose the move with the lowest value for you. When the call returns, you choose the highest value from the choices you've simulated for your opponent.

1. Cutoffs

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

- (a) ply

Solution:

The total number of levels ahead looked (so, the height of your search tree).

- (b) cutoff

Solution:

The number of levels remaining in the tree when your search switches to a non-parallel algorithm. Similar to the `sequentialCutoff` parameter from exercises.

- (c) divideCutoff

Solution:

The maximum number of threads which should be forked sequentially when dividing-and-conquering a list of moves.

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

Solution:

Create threads with divide-and-conquer.

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

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

Solution:

Copy the board *inside* the child thread.