

# An Alpha-Beta Primer

Stephen D. Bay

Minimax search with alpha-beta pruning involves depth first search of a game tree, keeping track of

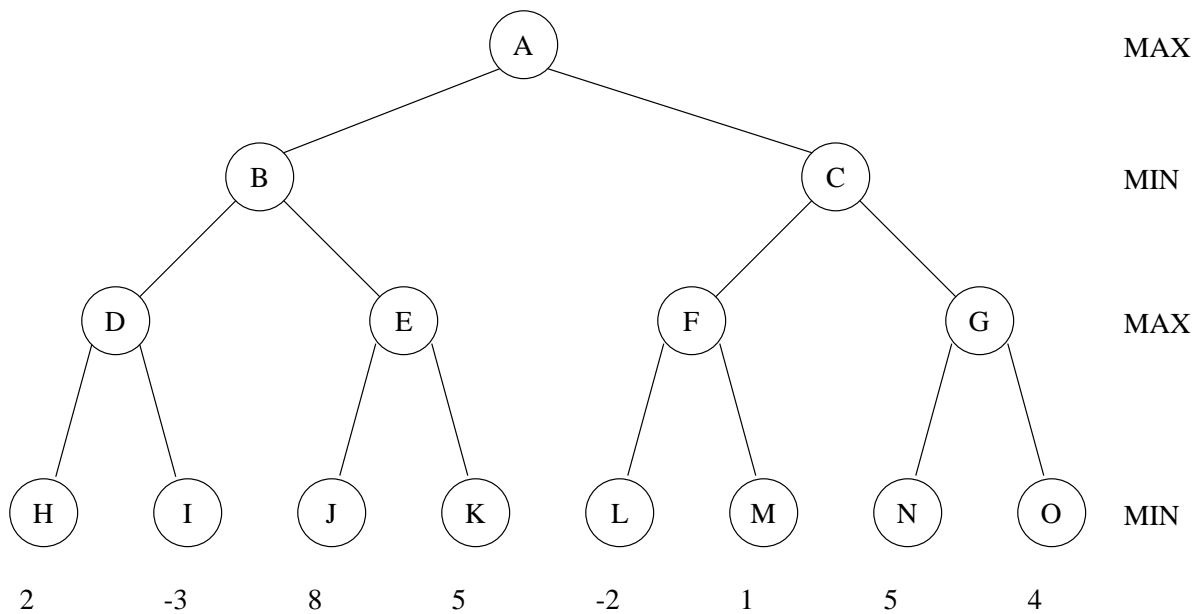
- Alpha: highest value seen so far on a maximizing level
- Beta: lowest value seen so far on a minimizing level

Note that when we look at alpha and beta values we look only at nodes along the path from our current node to the root.

Pruning is done according to the following two rules:

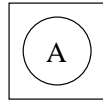
- When on a maximizing level, do not expand any more sibling nodes once a node has been seen whose evaluation is lower than or equal to Alpha.
- When on a minimizing level, do not expand any more sibling nodes once a node has been seen whose evaluation is greater than or equal to Beta.

For this primer, we will consider the following search tree:



This is the full search tree to depth 3. The numbers beneath the leaves are values returned by the evaluation function.

**Step 1**



MAX

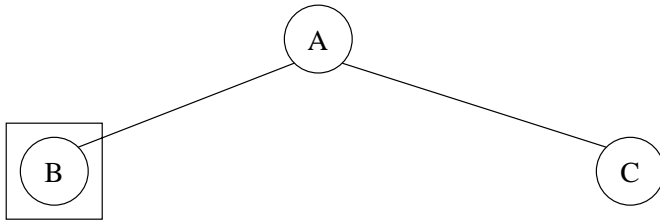
MIN

MAX

MIN

alpha = undefined, beta = undefined, Expand A

**Step 2**



MAX

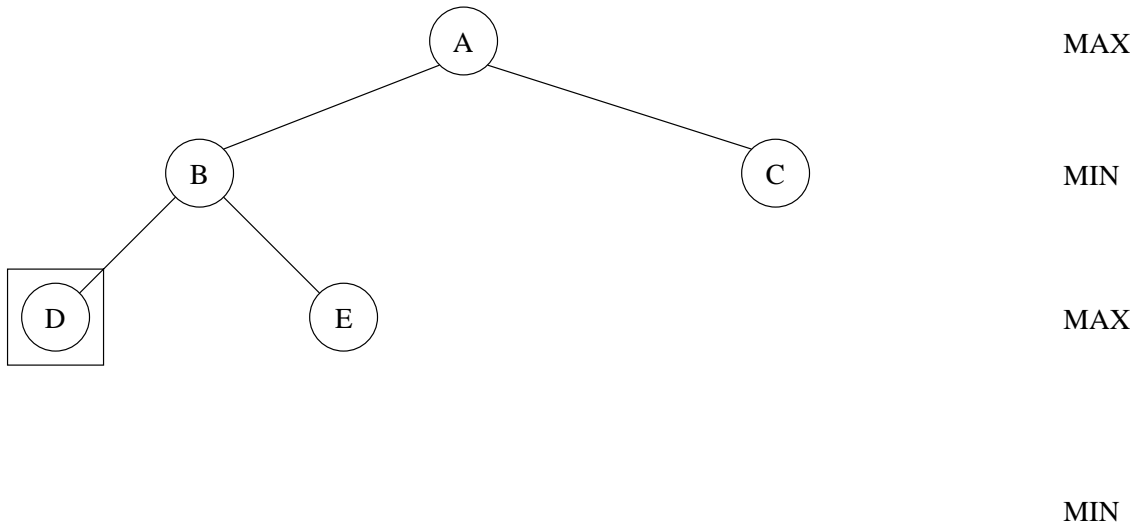
MIN

MAX

MIN

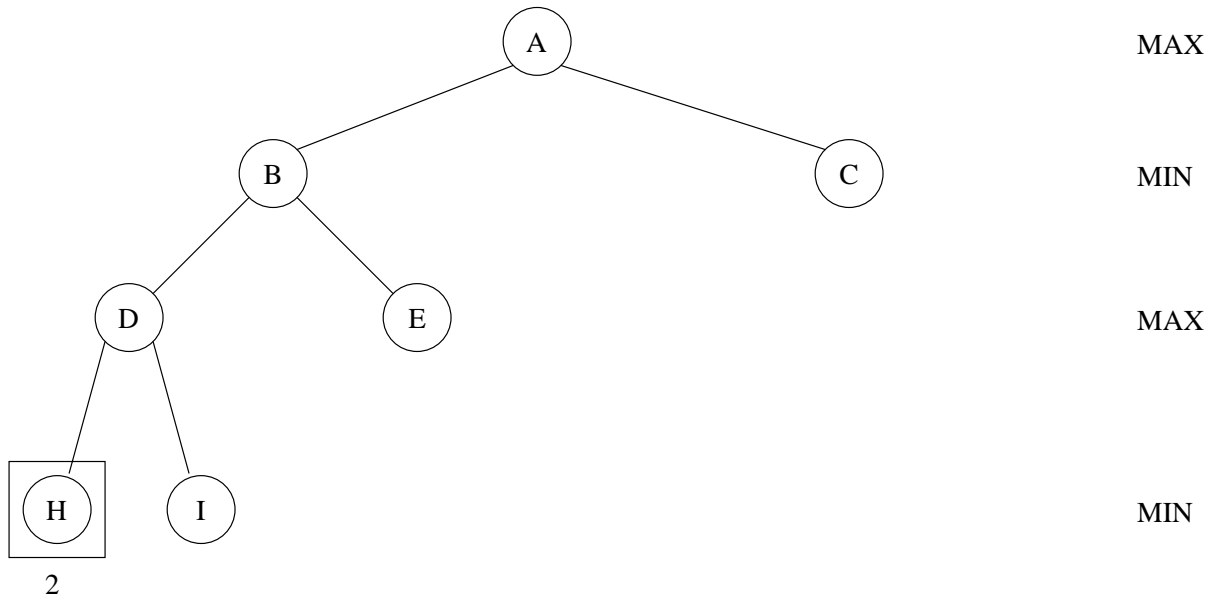
alpha = undefined, beta = undefined, Expand B

**Step 3**



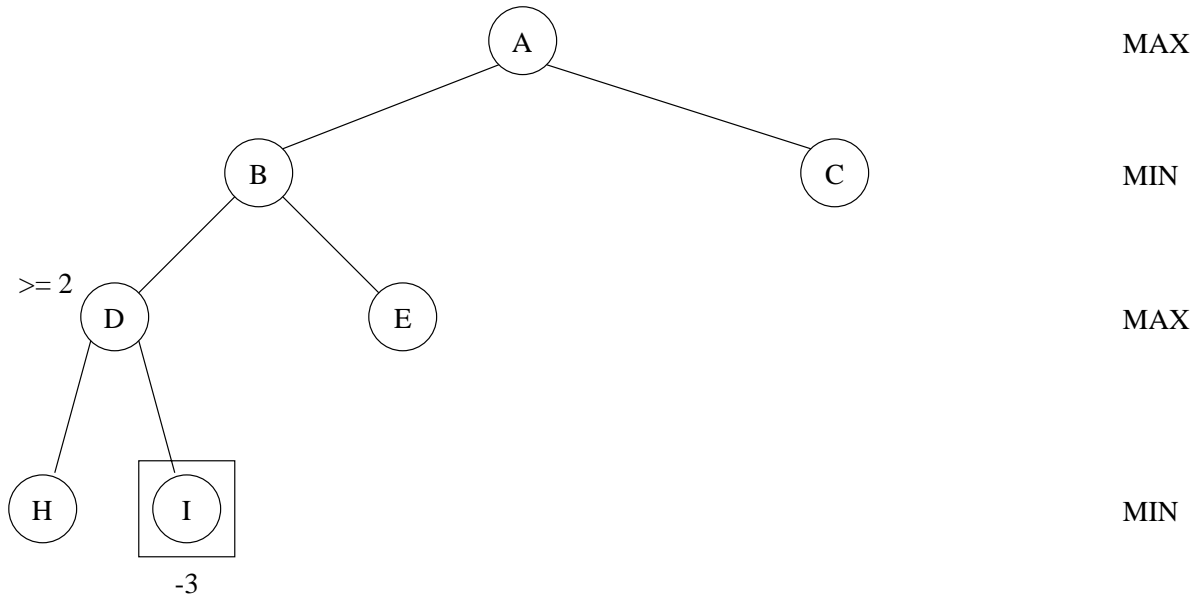
alpha = undefined, beta = undefined, Expand D

**Step 4**



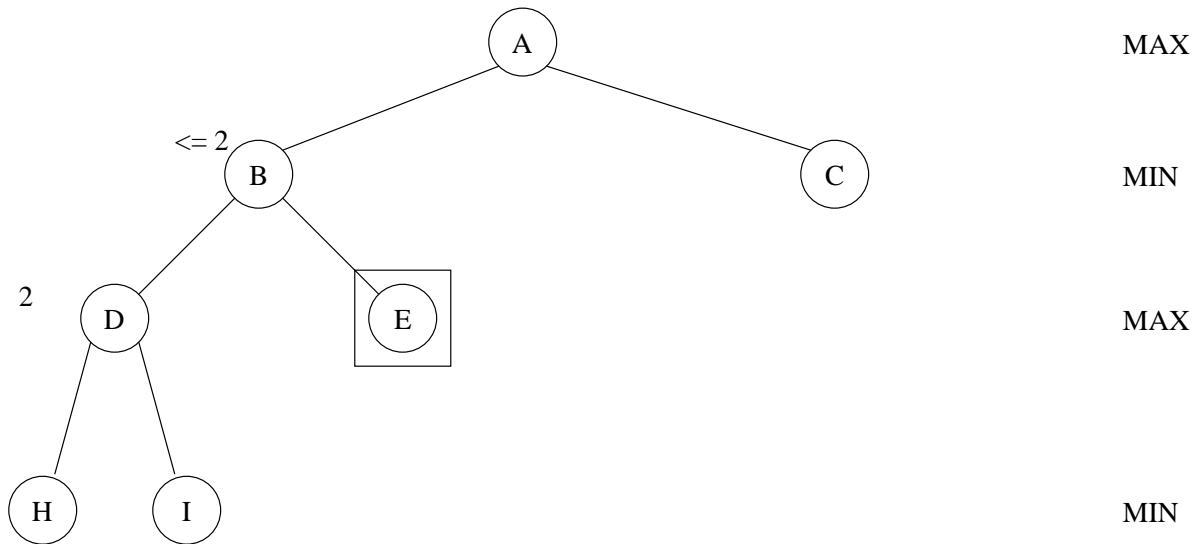
alpha = undefined, beta = undefined, Evaluate H and propagate values upwards.

Step 5



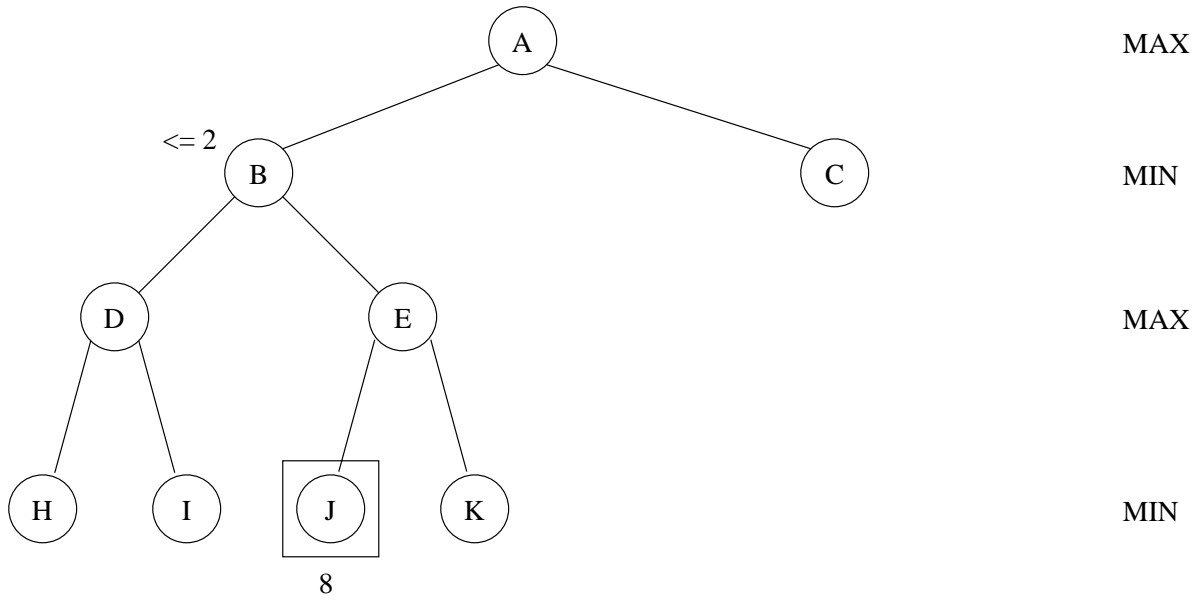
alpha = 2, beta = undefined, Evaluate I and propagate values upwards.

Step 6



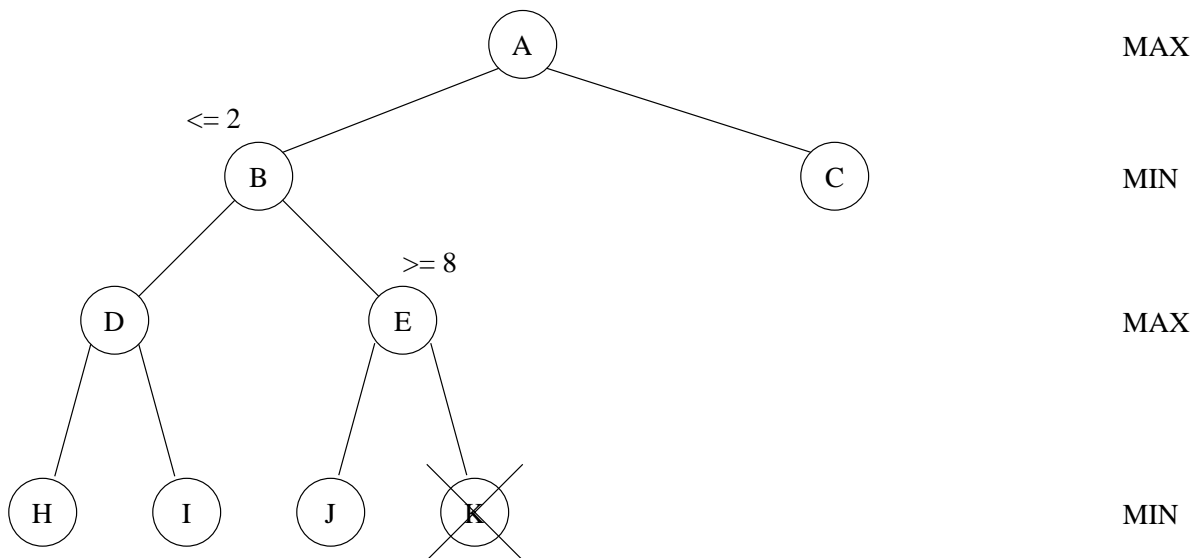
alpha = undefined, beta = 2, Expand E. Note that alpha is undefined since we are at E and we only consider alpha-beta values from E to the root A (i.e. values at B and A).

Step 7



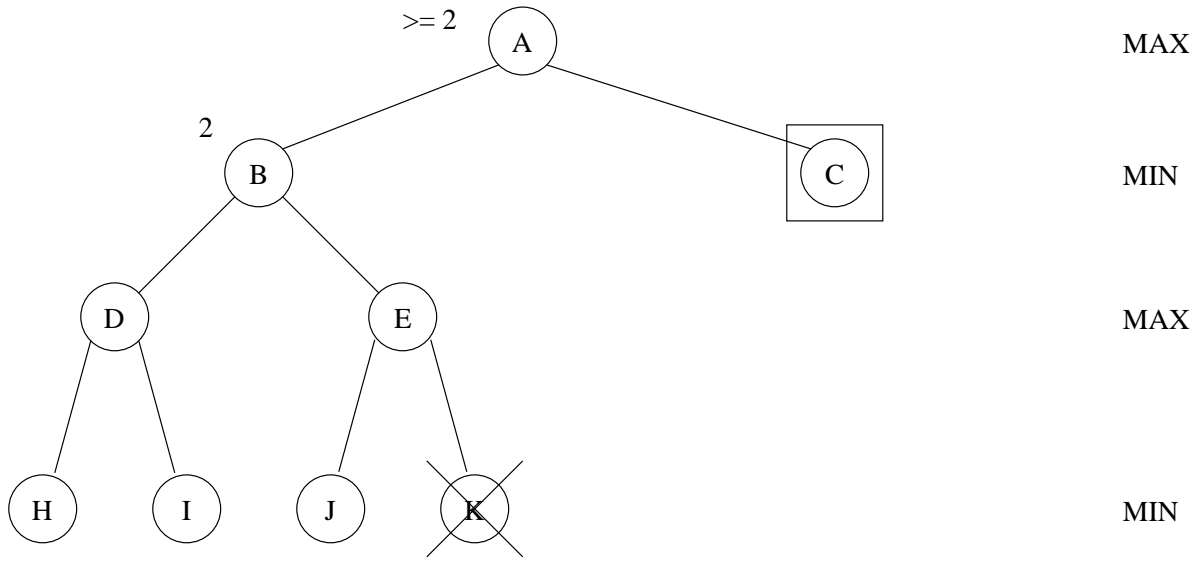
alpha = undefined, beta = 2, Evaluate J and propagate values upwards.

Step 8



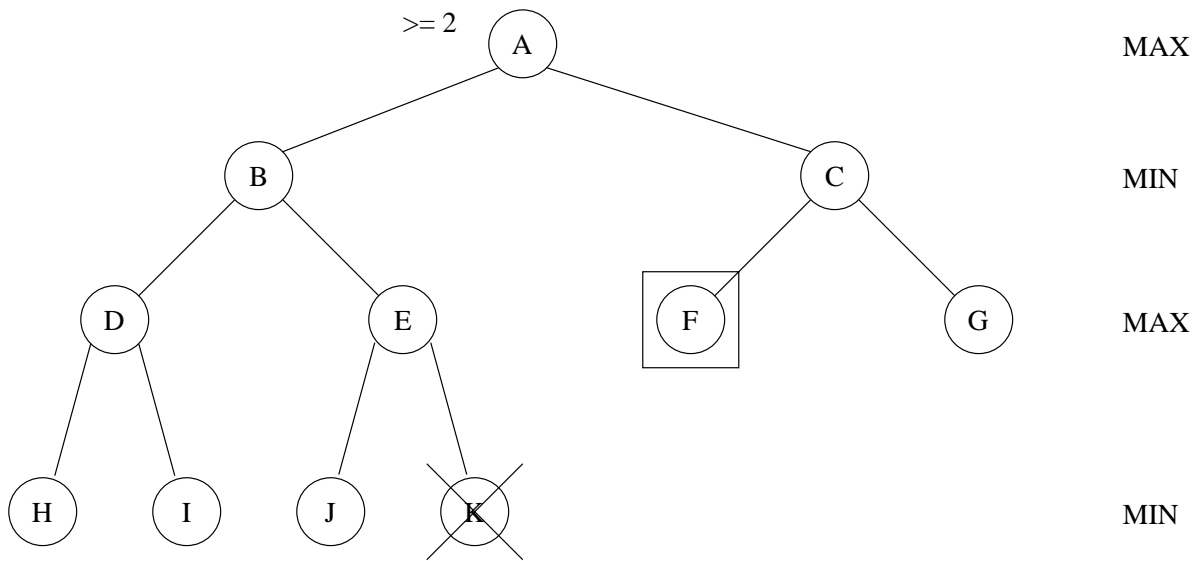
alpha = 8, beta = 2, Prune K as  $\text{eval}(j) \geq \beta$  ( $8 \geq 2$ ). Once K has been pruned we can propagate values up to A.

Step 9



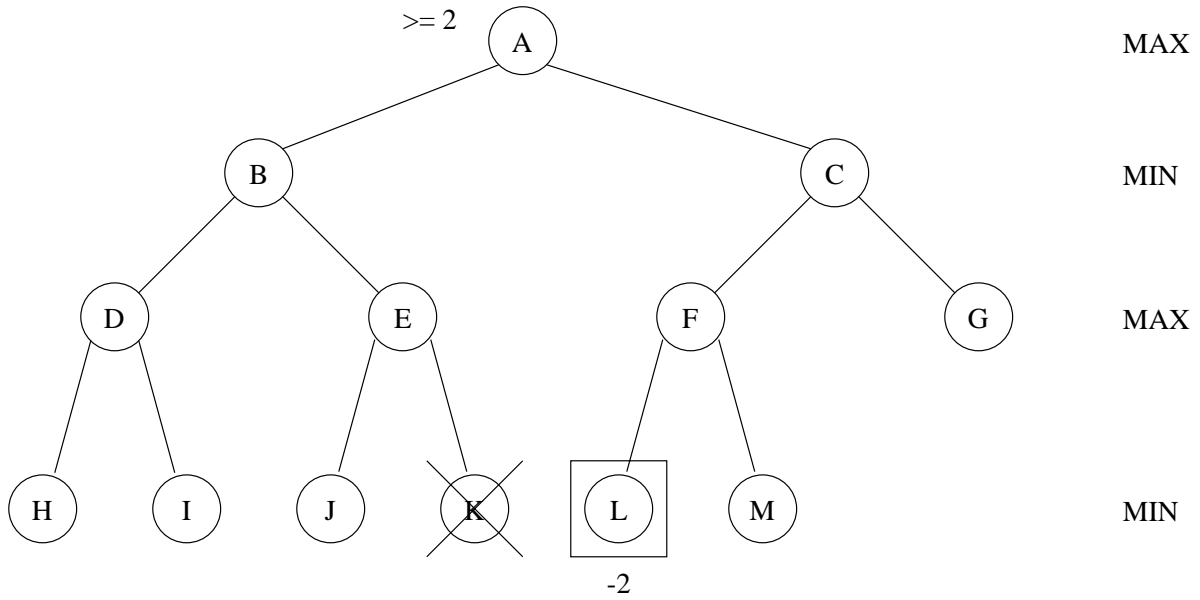
alpha = 2, beta = undefined, Expand C

Step 10



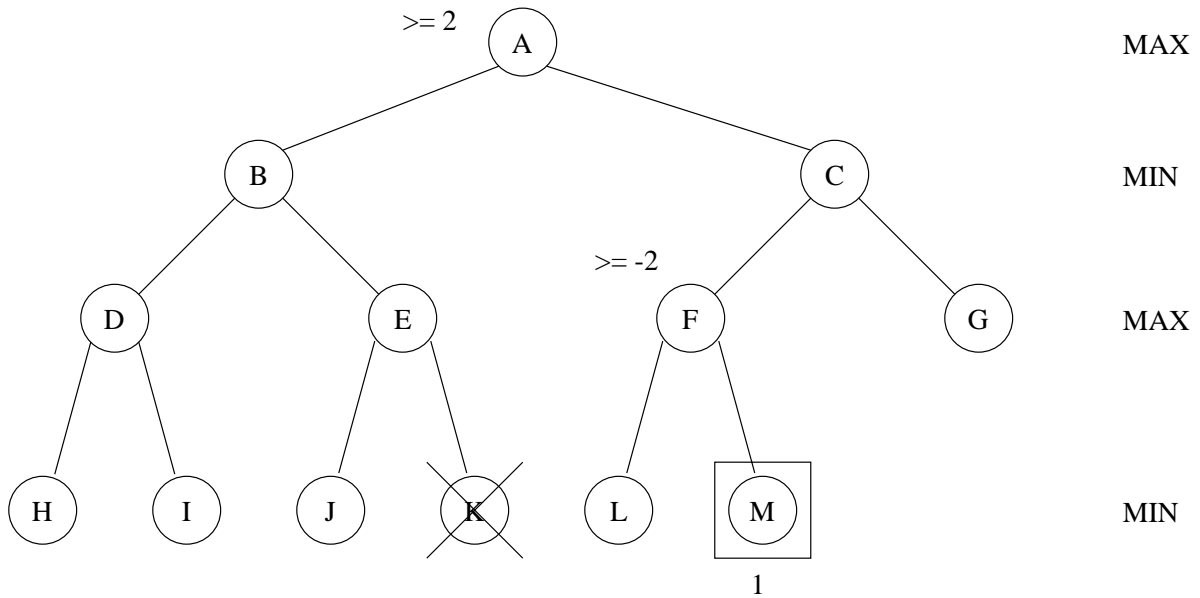
alpha = 2, beta = undefined, Expand F

Step 11



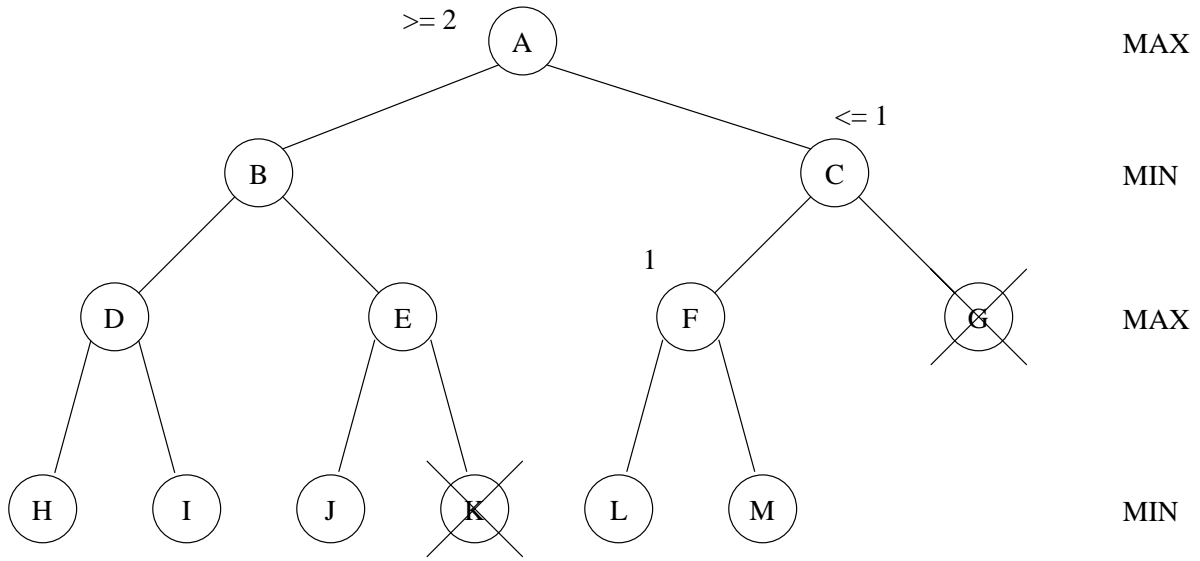
alpha = 2, beta = undefined, Evaluate L and propagate values upward.

Step 12



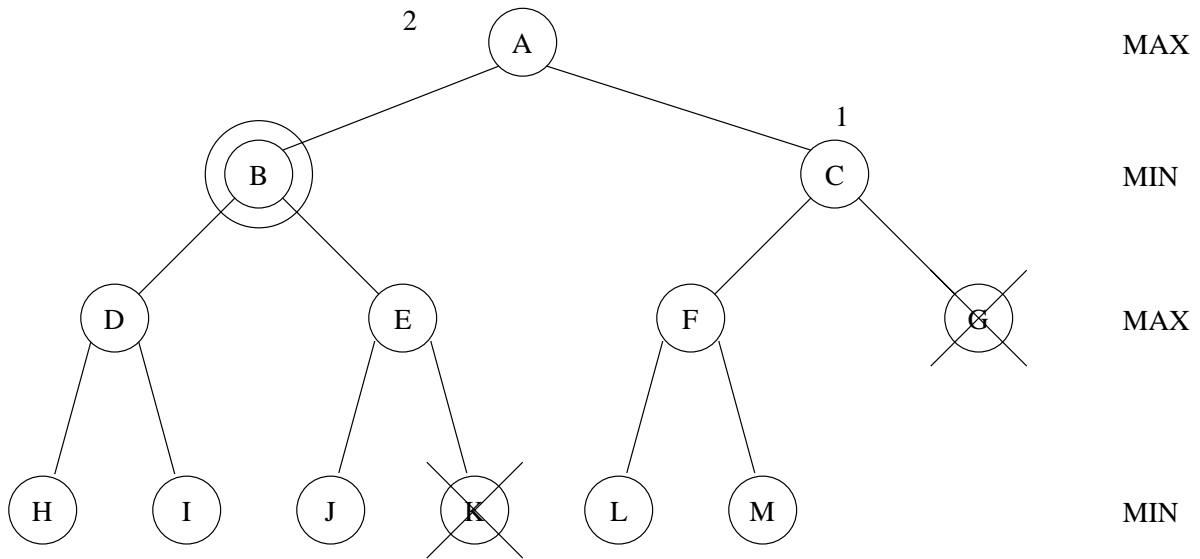
alpha = 2, beta = undefined, Evaluate M and propagate values upward.

Step 13



alpha = 2, beta = 1, Prune G as  $\text{eval}(F) \leq \alpha$  ( $1 \leq 2$ ). Once G has been pruned we can propagate values up to A.

Step 14



The computer chooses the move that leads to B.