Name: ID:

1. The Alpha-Beta pruning technique is used to improve the runtime of the minimax algorithm. With a constant branching factor of b, and a search depth of d, answer the following questions about it's performance:
   a. (2 point) What is the worst case runtime of minimax using Alpha-Beta pruning?

      $O(b^{d})$

   b. (2 point) What is the best case runtime of minimax using Alpha-Beta pruning?

      $O(b^{d/2})$

   c. (2 point) Under what conditions can we achieve the best case runtime?

      Optimal move ordering. Or more precisely
      for a max node, the first child has the largest value;
      for a min node, the first child has the smallest value.

   d. (2 point) Under what conditions will Alpha-Beta pruning not prune any branches at all?

      When the move ordering is pessimal, Alpha-Beta pruning is the same as a simple minimax search.
      for a max node, the children with smaller values are explored first;
      for a min node, the children with larger values are explored first.

function ALPHA-BETA-SEARCH(*state*) **returns** an action
  $v \leftarrow$ MAX-VALUE($state, -\infty, +\infty$)
  return the *action* in ACTIONS(*state*) with value v

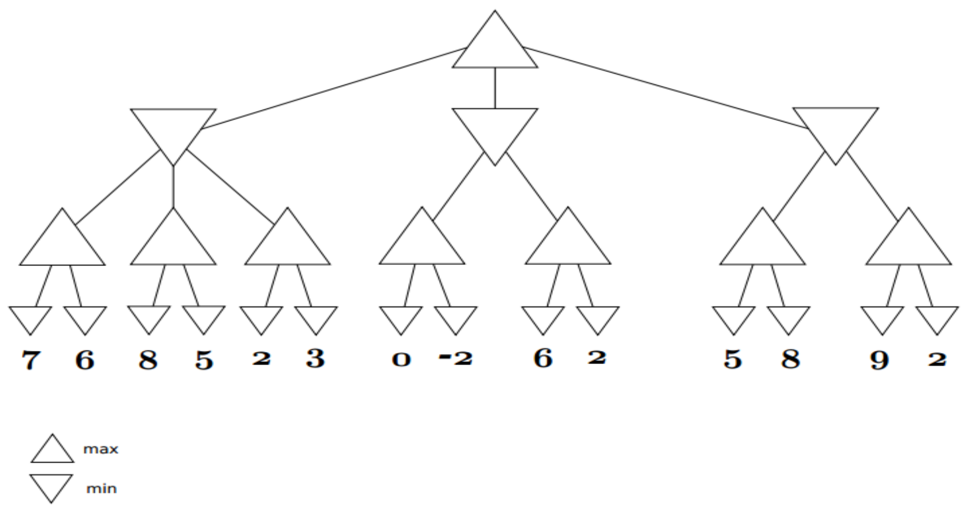
---

function MAX-VALUE($state, \alpha, \beta$) **returns** *a utility value*
  if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)
  $v \leftarrow -\infty$
  for each a **in** ACTIONS(*state*) **do**
    $v \leftarrow$ MAX(v, MIN-VALUE(RESULT(s,a), α, β))
    if $v \geq \beta$ **then return** v
    $\alpha \leftarrow$ MAX(α, v)
  return v

---

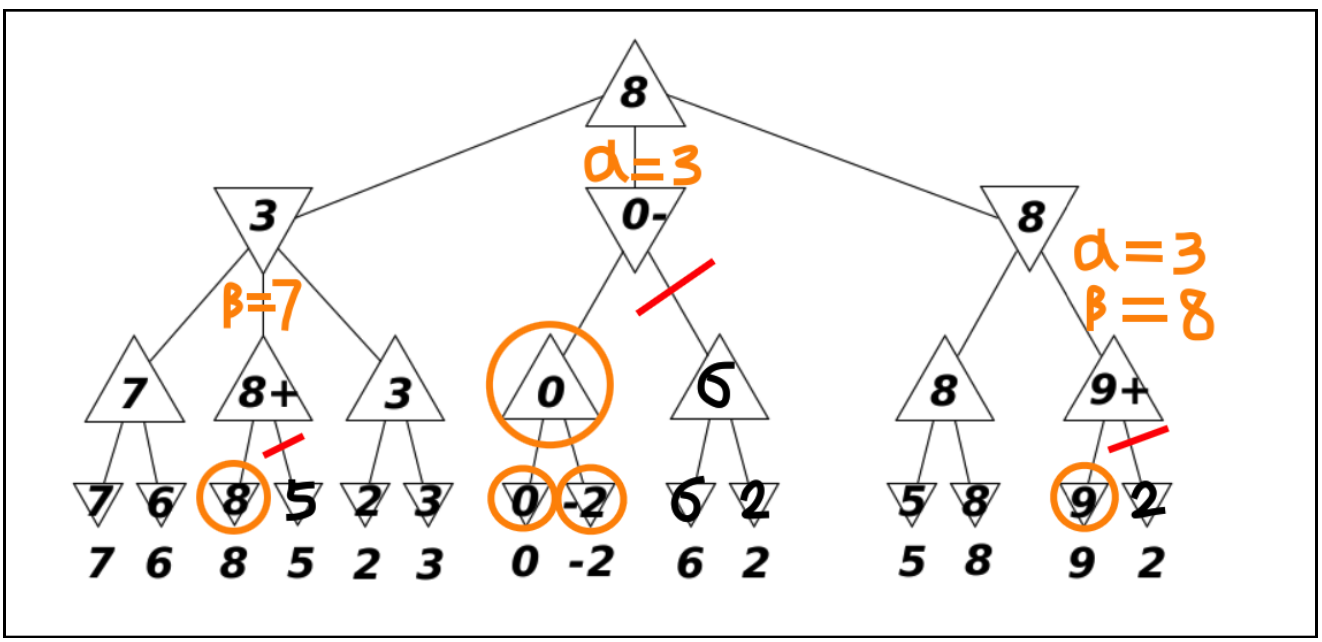
function MIN-VALUE($state, \alpha, \beta$) **returns** *a utility value*
  if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)
  $v \leftarrow +\infty$
  for each a **in** ACTIONS(*state*) **do**
    $v \leftarrow$ MIN(v, MAX-VALUE(RESULT(s,a) , α, β))
    if $v \leq \alpha$ **then return** v
    $\beta \leftarrow$ MIN(β, v)
  return v

Figure 5.7 The alpha–beta search algorithm. Notice that these routines are the same as the MINIMAX functions in Figure 5.3, except for the two lines in each of MIN-VALUE and MAX-VALUE that maintain α and β (and the bookkeeping to pass these parameters along).



2.

a. Execute alpha-beta pruning on the example, write the minimax value at each node (including the nodes got pruned), and cross out the branches that get pruned by the algorithm. If a branch does get pruned, circle the nodes under that branch that you had to explore in order to decide to prune the branch.



b. How would you reorder the first moves that max makes in order to improve alpha-beta pruning? (Hint: Reorder the subtrees of the root node)