```
function RECURSIVE-BEST-FIRST-SEARCH(problem) returns a solution or failure
  solution, fvalue ← RBFS(problem, NODE(problem.INITIAL), $\infty$)
 return solution

function RBFS(problem, node, f_limit) returns a solution or failure, and a new $f$-cost limit
  if problem.IS-GOAL(node.STATE) then return node
  successors ← LIST(EXPAND(node))
  if successors is empty then return failure, $\infty$
  for each s in successors do        // update f with value from previous search
      s.f ← max(s.PATH-COST + h(s), node.f)
  while true do
      best ← the node in successors with lowest $f$-value
      if best.f $>$ f_limit then return failure, best.f
      alternative ← the second-lowest $f$-value among successors
      result, best.f ← RBFS(problem, best, min(f_limit, alternative))
      if result $\neq$ failure then return result, best.f
```