Algorithm 1 Recursive Best First Search Algorithm

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



function RBFS(problem, node, f_limit) returns a solution, or failure and a new f-cost limit
    if $problem$.GOAL-TEST($node$.STATE) then return SOLUTION($node$)
    $successors \leftarrow [\,]$

    for each $action$ in $problem$.ACTIONS($node$.STATE) do
        $successors$.APPEND(CHILD-NODE($problem, node, action$))
    if $successors = \emptyset$ then return $failure, \infty$
    for each $successor$ in $successors$ do
        $successor.f \leftarrow \max(successor.g + successor.h, node.f)$
    loop do
        $best \leftarrow$ the lowest f-value node in successors
        if $best.f > f\_limit$ then return $failure, best.f$

        $alternative \leftarrow$ the second lowest f-value among successors
        $result,\ best.f \leftarrow$ RBFS($problem, best, min(f\_limit, alternative)$)

        if $result \neq failure$ then return $result$
```