●

QuickSort(A,First,Last)

var v,sp,L,L2,p,r,r2;   Mark the local
                               stack as empty.

sp=0;   Loads First and
               Last onto the stack

Push(First,Last);   «DecisionInput»
                           sp > 0

end while(sp>0)
end QuickSort

[No]

[Yes]

●

Pop(L, r)/2;

span is L:r   «DecisionInput»
                     L < r

[No]

[Yes]

end while(L < r)

the value must not   p = (L+r)/2;
stay in the array!

v = A[p].key;
L2 = L;
r2 = r;

«DecisionInput»   repeat until L2 >= r2
L2 < r2

[No]   [Yes]

is left side   «DecisionInput»   scan left partition   «DecisionInput»
piece larger?   r2-L > r-L2   A[L2].key < v

«DecisionInput»   [No]
L<r2

[Yes]   [No]   [Yes]

«DecisionInput»   if left side isn't, right   «DecisionInput»   scan right partition   L2 = L2+L;
L2 < r   side is larger   A[r2].key > v

[No]

[Yes]   «DecisionInput»
                                                                      L2 <= r2
Push(L,r2);

                                                                      [No]

[No]   [Yes]   [Yes]

Push(L2,r);
r = r2;

r2 = r2-L;

«DecisionInput»
L2 equals r2

[Yes]

[No]

[Yes]

[No]

L = L2;

Swap(A[L2],A[r2]);

L2 = L2 + L;
r2 = r2 - L;