

function vars={items}

entry

if (start+1) < end:

Yes

No

then

...

else

if-block-scope#1

outside vars={items}

in={}

**Set**  
gen={tmp, lhs}  
transfer={}  
kill={}

**Code Block**  
//Slice the array items  
int[] tmp = Array.slice(items,start,pivot)  
int[] lhs = clone(tmp)

out={tmp, lhs}

outside vars={items}

in={tmp, lhs}

**Set**  
gen={tmp1, lhs}  
transfer={}  
kill={lhs}

**Code Block**  
// Recursively sort the lhs array  
int[] tmp1 = sort(clone(lhs), 0, pivot)  
lhs = clone(tmp1)

out={tmp, tmp1, lhs}

outside vars={items}

in={tmp, tmp1, lhs}

**Set**  
gen={tmp2, rhs}  
transfer={}  
kill={}

**Code Block**  
//Slice the items array  
int[] tmp2 = Array.slice(items,start,pivot)  
int[] rhs = clone(tmp2)

out={tmp, tmp1, tmp2, lhs, rhs}

outside vars={items}

in={tmp, tmp1, tmp2, lhs, rhs}

**Set**  
gen={tmp3, rhs}  
transfer={}  
kill={rhs}

**Code Block**  
// Recursively sort the rhs array  
int[] tmp3 = sort(clone(rhs), 0, end-pivot)  
rhs = clone(tmp3)

out={tmp, tmp1, tmp2, tmp3, lhs, rhs}

while-loop#1

in={}

Use while-loop  
to merge lhs and rhs...

out={}

block vars=in={}

in={}

//No free vars  
Loop Exit

while-loop#2"

in={}

Use while-loop  
to merge remaining lhs...

out={}

block vars=in={}

in={}

//No free vars  
Loop Exit

while-loop#3

in={}

Use while-loop  
to merge remaining rhs...

out={}

block vars=in={}

in={}

//No free vars  
Loop Exit

block vars=in={tmp, tmp1, tmp2, tmp3, lhs, rhs}

free(tmp)  
free(tmp1)  
free(tmp2)  
free(tmp3)  
free(lhs)  
free(rhs)  
//End of if block

End if

function vars={items}

//No free variable  
return items  
//End of function

exit