1. Recall that the claim that the calls to determine L and R give the correct result, in the proof of correctness for mss3, requires that k1and k2 are < n-1 for n>2, where k1=m-p1+1, k2=p2-(m+1)+1, m=floor((p1+p2)/2), and n=p2-p1+1. Prove that k1<= n-1 and k2<= n-1 for n>=2.

1. k1=m-p1+1
2. k2=p2-(m+1)+1
3. m=floor((p1+p2)/2)
4. n=p2-p1+1

prove

* k1< n-1 and k2< n-1, for n>2.

Soluton:

Let us analyze length of left part. Putting 3) into 1) we get:

k1 = floor((p1+p2)/2) – p1 + 1 = floor((p1+p2)/2) – 2\*p1/2 + 1 = floor((p2 – p1)/2 + 1) ,

knowing conditions that p1 >= 1 and p2 <= n we can write:

k1 **<=** floor((n –1)/2 + 1) = floor((n + 1)/2) .

Now do the same thing for the length of right part. Putting 3) into 2) we get:

k2 = p2 - (floor((p1+p2)/2) + 1) + 1 = p2 – floor((p1+p2)/2) ,

again using same conditions as above, we can write:

k2 **<=** n - floor((1 + n)/2) = floor( - (n + 1)/2 + (2\*n)/2) = floor((n - 1) / 2) .

Now that k1 and k2 have been expressed as functions of n, we can continue with proof of k1 and k2 being <= n – 1, for all n >= 2. Because we know that k2 < k1, if the inequality k1 <= n – 1 is satisfied, n2 will be <= n – 1, so it is enough to prove inequality k1 <= n – 1.

We must split proof into two parts:

n is odd, or n is even.

* If n is even, then n = 2\*k, k>=1 (definition of even numbers, k must be an integer). Considering this we now have:

k1 <= floor((n + 1)/2) = floor((2\*k + 1)/2) = floor(k + 1/2) = k,

as k = n/2:

k1 <= floor(n/2), therefore k1 must be less or equal to n – 1, if n is greater or equal than 2 (remember that k1 must be and integer). And as it is shown above, k2 will be also <= n – 1, n >= 2.

* If n is odd, then n = 2\*k + 1, k>=1 (definition of odd numbers, k must be an integer). From this we have:

k1 <= floor((n + 1) / 2) = floor((2\*k + 1 + 1) / 2) = floor(k+1), as k = (n – 1)/2:

k1 <= floor(1 + (n - 1)/2) = floor((n + 1)/2), therefore k1 must be less or equal to n -1, for all n >= 2 (remember that k1 must be and integer). And as it is shown above, k2 will be also <= n – 1, n >= 2.

3. Find a loop invariant that works for this modified mss4 and show it is “useful”. Do not prove it is a loop invariant.

For loop to be useful !C and L must imply the correct result. For the result correctness result be correct when program stops, and program must stop.

!C: i is n+1

L: