Alg separatePosNNeg(A[0…..n-1])

for i 0 to n-1

do if (A[i]>0 && A[i+1]<0)

then tmp=A[i] for loop:

A[i]=A[i+1] = n-1-0+1 = n

A[i+1]= tmp Ꝋ(n)

i=i-2

Func separatePosNNeg(A, l, r){

if(l==r){

return l; c/1

}

else{

separatePosNNeg(A, l, floor(l+r)/2) T(n/2)

separatePosNNeg(A, floor(l+r)/2+1, r) T(n/2)

for i 0 to n-1

do if (A[i]>0 && A[i+1]<0){

tmp=A[i] for loop:

A[i]=A[i+1] =

A[i+1]= tmp n-1-0+1 = n

i=i-2

}

}

If\_else = max(c,2T(n/2)+n)= 2T(n/2)+n (master method)

nlog22 = n Ꝋ(n)

**By comparison:**

time complexity of non-recursive algorithm(Ꝋ(n))

= time complexity of recursive algorithm(Ꝋ(n))