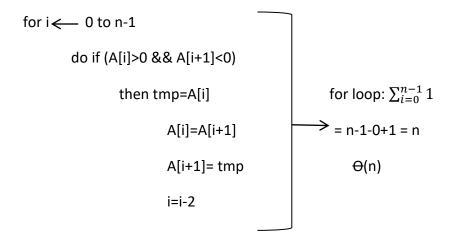
Alg separatePosNNeg(A[0.....n-1]) // non\_recursive



Func separatePosNNeg(A, I, r){ // recursive

$$if(l==r)\{ \\ return \ l; \longrightarrow c/1 \\ \} \\ else\{ \\ separatePosNNeg(A, l, floor(l+r)/2) \longrightarrow T(n/2) \\ separatePosNNeg(A, floor(l+r)/2+1, r) \longrightarrow T(n/2) \\ for \ i \longleftarrow 0 \ to \ n-1 \\ do \ if \ (A[i]>0 \ \&\& \ A[i+1]< \\ tmp=A[i] \\ A[i]=A[i+1] \\ A[i+1]=tmp \\ i=i-2 \\ \}$$
 
$$for \ loop: \ \sum_{i=0}^{n-1} 1 = n$$

```
}  If_else = max(c,2T(n/2)+n) = 2T(n/2)+n \text{ (master method)}   n^{\log_2 2} = n \quad \underline{\hspace{1cm}} (n)
```

## By comparison:

time complexity of non-recursive algorithm( $\Theta(n)$ )

= time complexity of recursive algorithm( $\Theta(n)$ )