

Dato il seguente algoritmo di ordinamento:

STOOGESORT(A, i, j)

```
1  if  $A[i] > A[j]$ 
2    then scambia  $A[i] \leftrightarrow A[j]$ 
3  if  $i + 1 \geq j$ 
4    then return
5   $k \leftarrow \text{floor}((j - i + 1) / 3)$       ► Suddivisione.
6  STOOGESORT( $A, i, j - k$ )              ► Primi due terzi.
7  STOOGESORT( $A, i + k, j$ )              ► Ultimi due terzi.
8  STOOGESORT( $A, i, j - k$ )              ► Ancora i primi due terzi.
```

- 1) scrivere e risolvere la ricorrenza che descrive $T(n)$, con $n = \text{length}[A]$, e confrontare il risultato trovato con la complessità nel caso peggiore di *Insertion Sort* e *Quicksort*.
- 2) dimostrare la correttezza dell'algoritmo, ovvero che la chiamata $\text{STOOGESORT}(A, 1, \text{length}[A])$ ordina correttamente l'array di input $A[1 \dots n]$.