

Pseudocodice

D. Quicksort

D.1. Quicksort: schema ad alto livello

Algoritmo *quickSort* (array A)
if *lunghezza di* $A > 1$ **then**
 scegli un elemento di A come *perno*
 $B \leftarrow \{y \in A \mid y \leq \textit{perno}\}$
 $C \leftarrow \{y \in A \mid y > \textit{perno}\}$
 quickSort(B)
 quickSort(C)
 $A \leftarrow$ concatenazione di B e C

D.2. Partizionamento di un array

Algoritmo *partiziona* (array A , indice i , indice f) \rightarrow indice
/* Riorganizza gli elementi all'interno di $A[i..f-1]$ e restituisce un
 indice j in modo tale che tutti gli elementi di $A[i..j-1]$ siano minori o
 uguali ad $A[j]$ e tutti gli elementi di $A[j+1..f-1]$ siano maggiori di $A[j]$
*/
 $\textit{perno} \leftarrow A[i]$
 $sx \leftarrow i$
 $dx \leftarrow f$
while $sx < dx$ **do**
 do $dx \leftarrow dx - 1$ **while** $A[dx] > \textit{perno}$
 do $sx \leftarrow sx + 1$ **while** $sx < dx$ **and** $A[sx] \leq \textit{perno}$
 if $sx < dx$ **then** scambia $A[sx]$ con $A[dx]$
scambia $A[i]$ con $A[dx]$
return dx

D.3. Quicksort

Algoritmo *quickSort* (array $A[0..n - 1]$)
quickSort($A, 0, n$)

Procedura *quickSort* (array A , indice i , indice f) /* Ordina $A[i..f - 1]$ */
if $f - i > 1$ **then**
 $m \leftarrow \textit{partiziona}(A, i, f)$
 quickSort(A, i, m)
 quickSort($A, m + 1, f$)

D.4. Procedura ricorsiva di Quicksort senza ricorsione in coda

Procedura *quickSort* (array A , indice i , indice f) /* Ordina $A[i..f - 1]$ */
while $f - i > 1$ **do**
 $m \leftarrow \textit{partiziona}(A, i, f)$
 quickSort(A, i, m)
 $i \leftarrow m + 1$

D.5. Procedura ricorsiva di Quicksort con riduzione dell'altezza dello stack

Dopo il partizionamento, ordina subito la parte pi\`u piccola ottenuta (chiamata ricorsiva), lasciando "in sospenso" (per la successiva iterazione del ciclo) la parte pi\`u grande. In tal modo, ogni record di attivazione sullo stack rappresenta una chiamata per ordinare un array di lunghezza al pi\`u met\`a rispetto a quella dell'array del record del chiamante. Di conseguenza l'altezza dello stack risulta \`e al massimo dell'ordine di $\log n$.

Procedura *quickSort* (array A , indice i , indice f) /* Ordina $A[i..f - 1]$ */
while $f - i > 1$ **do**
 $m \leftarrow \textit{partiziona}(A, i, f)$
 if $m - i < f - m$ **then**
 quickSort(A, i, m)
 $i \leftarrow m + 1$
 else
 quickSort($A, m + 1, f$)
 $f \leftarrow m$