

Counting Sort

Algoritmo d'ordinamento in tempo lineare

COUNTING SORT (A, m)

$(\text{Min}, \text{Max}) \leftarrow \text{FIND_MIN_MAX}(A, m)$

$K \leftarrow \text{max} - \text{min} + 1$

$C \leftarrow \text{NEW_ARRAY}(K)$

fase di
inizializzazione
dell'array

FOR $i \leftarrow 0$ TO $K-1$ DO

$C[i] \leftarrow 0$

inizializzo a 0
TUTTI i VALORI

FOR $i \leftarrow 0$ TO $m-1$ DO

$C[A[i] - \text{min}] \leftarrow C[A[i] - \text{min}] + 1$

$B \leftarrow \text{NEW_ARRAY}[m]$

NUOVO ARRAY

$J \leftarrow 0$

FOR $i \leftarrow 0$ TO $K-1$ DO

FOR $h \leftarrow 1$ TO $C[i]$ DO

$B[J] \leftarrow i + \text{min}$

$J \leftarrow J + 1$

Si va
sommando il
min in maniera
ordinata

Come possiamo notare l'algoritmo non lavora in loco e presenta molte limitazioni

Ne tiene a lavorare con un tempo
 $O(K+m)$ che è di tipo lineare

Versione Counting Sort

COUNTING SORT(A, m)

$(min, max) \leftarrow \text{Find } min \text{ } max(A, m)$

$K \leftarrow max - min + 1$

$C \leftarrow \text{NEW ARRAY } [K]$

FOR $i \leftarrow 0$ TO $K-1$ DO

$C[i] \leftarrow 0$

FOR $i \leftarrow 0$ TO $m-1$ DO

$C[A[i] - min] \leftarrow C[A[i] - min] + 1$

FOR $i \leftarrow 1$ TO $K-1$ DO

$C[i] \leftarrow C[i] + C[i-1]$

$B \leftarrow \text{NEW ARRAY } [m]$

FOR $i \leftarrow m-1$ DOWN TO 0 DO

$B[C[A[i] - min] - 1] \leftarrow A[i]$

$C[A[i] - min] \leftarrow C[A[i] - min] - 1$

Si usa questo codice che partendo da $n-1$, invece che da 0 tiene e preserva la stabilità dell'ordinamento

Il nuovo ciclo for introdotto somma tutti gli elementi che potenzialmente precedono $C[i]$ nell'array in modo da avere un controllo utile per mantenere la stabilità.