

Counting Sort

Conto le occorrenze di tutti i numeri nella mia sequenza

Input:\

- sequenza di interi
 $A = A[0] \dots A[n-1]$
- $k \in \mathbb{N}$
 $\forall i \in [0, n - 1]$
 $0 \leq A[i] \leq k$

Output: A ordinata in senso non decrescente.

```
COUNTINGSORT_V1(A, k)
  n := A.length()
  crea C[0...k]
  for i = 0 to k    //inizializzo C
    C[i] := 0
  for j = 0 to n - 1
    C[A[j]] := C[A[j]] + 1
  j := 0
  for i = 0 to k
    while C[i] > 0
      A[j] := i
      j := j + 1
      C[i] := C[i] - 1
```

Questo algoritmo non ordina facendo confronti, non abbiamo l'ipotesi secondo il quale l'algoritmo ordina utilizzando confronti

STABILI: n^i uguali si trovano nello stesso ordine nella sequenza finale

Questa versione per esempio non è stabile, ma può diventarlo

Versione Stabile

1. Creo $C[i] = \text{n° occorrenze di } i$

2. $C[i] = \text{n° valori in } A \leq i$

3. $A \Rightarrow B$ "al posto giusto"

Per sapere la posizione nell'array ordinato devo conoscere tutti i valori minori o uguali

```
COUNTINGSORT_V2(A, k)
```

```
·
```

```
·
```

```
· (uguale fino a riga 6)
```

```
for i = 1 to k
```

```
    C[i] := C[i] + c[i - 1]
```

```
for j = n-1 down to 0
```

```
    B[C[A[j]] - 1] := A[j] - 1
```

```
Copia B in A
```

Se prima lo spazio era un $O(k)$ ora è un $O(k + n)$, lo spazio in più è lineare in base ad n , il tempo è lo stesso.