

ESAME DI ALGORITMI E STRUTTURE DI DATI 1
27 Giugno 2006

Esercizio 1 (Punti 15)

Sia A un vettore di numeri reali. Scrivere una procedura $\text{MaxRepeat}(A)$ che restituisce l'elemento di A che si ripete più volte in A . Valutare la complessità pessima della procedura scritta.

Soluzione

```
1: MaxRepeat( $A$ )
2: HeapSort( $A$ )
3:  $max \leftarrow 0$ 
4:  $count \leftarrow 1$ 
5: for  $i \leftarrow 1$  to  $length(A)$  do
6:   if  $i < length(A)$  and  $A[i] = A[i + 1]$  then
7:      $count \leftarrow count + 1$ 
8:   else
9:     if  $count > max$  then
10:       $max \leftarrow count$ 
11:       $elmax \leftarrow A[i]$ 
12:     end if
13:      $count \leftarrow 1$ 
14:   end if
15: end for
16: return  $elmax$ 
```

La complessità pessima risulta $O(n \log n)$, ove $n = length(A)$.

Esercizio 2 (Punti 15)

Sia $n \geq 1$ un intero e A un vettore di lunghezza n tale che, per ogni indice i di A vale $A[i] \in \{1, 2, \dots, n\}$. Scrivere una procedura $\text{MaxRepeat}(A)$ che restituisce l'elemento di A che si ripete più volte in A . La procedura **deve** avere complessità pessima $O(n)$.

Soluzione

Uso un vettore di appoggio C di lunghezza n .

```
1: MaxRepeat( $A$ )
2: for  $i \leftarrow 1$  to  $n$  do
3:    $C[i] \leftarrow 0$ 
4: end for
5: for  $i \leftarrow 1$  to  $n$  do
6:    $C[A[i]] \leftarrow C[A[i]] + 1$ 
7: end for
8: return ArrayMax( $C$ )
```