ESERCIZIO DI ASD DEL 6 OTTOBRE 2008

SELECTION-SORT

Algorithm 1 SELECTIONSORT(A)

```
for i ← 1 to length(A) − 1 do
   minpos ← i
   for j ← i + 1 to length(A) do
      if A[j] < A[minpos] then
         minpos ← j
      end if
   end for
   SCAMBIA(A, i, minpos)
end for
```

Algorithm 2 SCAMBIA(A, p, q)

```
x ← A[p]
A[p] ← A[q]
A[q] ← x
```

Correttezza.

Invariante 1. *All'inizio della j-esima iterazione del ciclo for più interno $A[minpos]$ è minore o uguale di ogni elemento in $A[i..j-1]$, ovvero*

$$\forall k \in [i..j-1](A[minpos] \leq A[k])$$

Nota: nella dimostrazione useremo la notazione $A[minpos] \leq A[i..j-1]$ per indicare che $A[minpos]$ è minore o uguale di ogni elemento in $A[i..j-1]$.

Dimostrazione. Per induzione su j.
Base: $j = i + 1$.
All'inizio dell'iterazione in cui j vale $i+1$ si ha che $minpos = i$ e $A[i..j-1] = A[i]$. Vale banalmente che $A[minpos] = A[i] \leq A[i]$.
Passo:
HpInd) All'inizio della $(k-1)$-esima iterazione del ciclo for vale che $A[minpos] \leq A[i..k-2]$.

Date: 6 Ottobre 2008.

TsInd) All'inizio della k-esima iterazione del ciclo for vale che $A[minpos] \leq A[i..k-1]$.

Tra l'inizio della $(k-1)$-esima iterazione e l'inizio della k-esima iterazione viene eseguita la $(k-1)$-esima iterazione del ciclo for. Quindi dobbiamo esaminare cosa avviene durante tale iterazione. La riga di codice 4 confronta $A[k-1]$ con $A[minpos]$. Se $A[k-1]$ è minore di $A[minpos]$, la riga 5 assegna a $minpos$ il valore $k-1$. Per ipotesi induttiva $A[minpos]$ era il più piccolo in $A[i..k-2]$. Se $A[k-1]$ è maggiore o uguale di $A[minpos]$, allora la riga 5 non viene eseguita e vale che $A[minpos] \leq A[i..k-1]$. Se $A[k-1]$ è minore di $A[minpos]$, allora vale che $A[k-1] < A[minpos] \leq A[i..k-2]$ ed inoltre alla riga 5 la variabile $minpos$ assume come nuovo valore $k-1$. Quindi dopo l'esecuzione della riga 5 vale che $A[minpos] = A[k-1] \leq A[i..k-1]$. □

L'invariante ci dice che all'inizio dell'iterazione in cui j assume valore $length(A)+1$ vale che $A[minpos] \leq A[i..length(A)]$. Inoltre all'inizio di quest'iterazione la condizione $j \leq length(A)$ che si trova a guardia del ciclo for non è più verificata ed il ciclo termina.

Invariante 2. *All'inizio della i-esima iterazione del ciclo for più esterno vale che:*
- $A[1..i-1]$ *è ordinato;*
- $A[1..i-1] \leq A[i..length(A)]$.

Dimostrazione. Per induzione su i.
Base: $i = 1$

$A[1..0]$ è un vettore vuoto, quindi è banalmente ordinato. Vale anche banalmente che $A[1..0] \leq A[1..length(A)]$, sempre perchè il primo non contiene elementi.
Passo:

HpInd) All'inizio dell'iterazione $k-1$ del ciclo for vale che $A[1..k-2]$ è ordinato e $A[1..k-2] \leq A[k-1..length(A)]$.

TsInd) All'inizio dell'iterazione k del ciclo for vale che $A[1..k-1]$ è ordinato e $A[1..k-1] \leq A[k..length(A)]$.

Dobbiamo esaminare cosa avviene durante l'iterazione $k-1$ del ciclo for. Per quanto dimostrato nell'Invariante precedente sappiamo che quando raggiungiamo la riga 8 del codice vale che $A[minpos] \leq A[k-1..length(A)]$. A questo punto viene eseguita la riga 8 che porta il $A[minpos]$ in posizione $k-1$. Quindi dopo l'esecuzione della riga 8 abbiamo che $A[k-1] \leq A[k..length(A)]$. Inoltre per ipotesi induttiva sappiamo che $A[1..k-2] \leq A[k-1]$. Riassumendo abbiamo che per ipotesi induttiva $A[1..k-2]$ era un vettore ordinato e non è stato modificato, inoltre sempre dall'ipotesi induttiva $A[1..k-2] \leq A[k-1]$ ed abbiamo dimostrato che $A[k-1] \leq A[k..length(A)]$. Quindi otteniamo che $A[1..k-1]$ è ordinato e $A[1..k-1] \leq A[k..length(A)]$. □

L'invariante ci dice che all'inizio dell'iterazione in cui i assume valore $length(A)$ vale che $A[1..length(A)-1]$ è ordinato e $A[1..length(A)-1] \leq A[length(A)]$. Inoltre all'inizio di quest'iterazione la guardia del ciclo for non è più soddisfatta ed il ciclo for termina.

Theorem 1. SELECTIONSORT(A) *termina sempre ed al termine il vettore A è ordinato.*

Dimostrazione. SELECTIONSORT(A) è costituito da un ciclo for, quindi termina quando termina il ciclo for. Il ciclo for contiene al suo interno un'altro ciclo for.

Il ciclo for più interno termina sempre in quanto viene eseguito al più $length(A)$ volte. Il ciclo più esterno termina per la stessa ragione. Quindi SELECTIONSORT(A) termina sempre.

Dall'Invariante precedente abbiamo che, al termine dell'esecuzione del ciclo for più esterno, $A[1..length(A) - 1]$ è ordinato e $A[1..length(A) - 1] \leq A[length(A)]$. Da questo segue immediatamente che al termine dell'esecuzione del for (e quindi di tutta la procedura) il vettore A è ordinato. $\square$

Complessità. Indichiamo con n la lunghezza di A. Analizziamo la complessità di ogni riga di codice:

1: Ha complessità $\Theta(1)$ e viene eseguita $\Theta(n)$ volte (per la precisione n volte);
2: Ha complessità $\Theta(1)$ e viene eseguita $\Theta(n)$ volte;
3: Ha complessità $\Theta(1)$ ed viene eseguita $\Theta(n - i)$ volte (per la precisione $n - i + 1$ volte) durante la i-esima iterazione del for esterno;
4: Ha complessità $\Theta(1)$ ed viene eseguita $\Theta(n-i)$ volte (per la precisione $n-i$ volte) durante la i-esima iterazione del for esterno;
5: Ha complessità $\Theta(1)$ ed viene eseguita $O(n-i)$ volte (per la precisione viene eseguita al più $n - i$ volte) durante la i-esima iterazione del for esterno;
6: Analogo alla riga 5;
7: Analogo alla riga 3;
8: Ha complessità $\Theta(1)$ e viene eseguita $\Theta(n)$ volte.

Sommando tutto ciò otteniamo

$$\begin{array}{rcl} T(n) & = & \Theta(n) + \sum_{i=1}^{n} -1\Theta(n-i) \\ & = & \Theta(n) + \sum_{i=1}^{n} -1c * (n-i) \\ & = & \Theta(n) + \sum_{i=1}^{n} -1c * n - \sum_{i=1}^{n} -1c * i \\ & = & \Theta(n) + c * n * (n-1) - c * \frac{n*(n-1)}{2} \\ & = & \Theta(n) + c * \frac{n*(n-1)}{2} \\ & = & \Theta(n^2) \end{array}$$