Insertion Sort

Wed, 16 Mar

L'algoritmo di Insertion Sort è un algoritmo di ordinamento.

Istanza d'esempio

```
▼ Insertion Sort applicato ad un array di esempio
```

```
• 72
29, 72
• 29, 72
                        ← 27
29, 27, 72
27, 29, 72
• 27, 29, 72
                        ← 15
27, 29, 15, 72

    27, 15, 29, 72

15, 27, 29, 72
15, 27, 29, 72
                        ← 78
15, 27, 29, 72, 78
15, 27, 29, 72, 78
                        ← 29
• 15, 27, 29, 72, 29, 78
• 15, 27, 29, <mark>29</mark>, 72, 78
• 15, 27, 29, 29, 72, 78
```

Valutazione tempi di esecuzione

▼ Pseudocodice dell'algoritmo

```
void InsertionSort(V[])
begin
                                                     c1 · (n-1)
   for j = 2 to n
   begin
      buff = v[j]
                                                     c2 · (n-1)
      i = j-1
                                                     c3 · (n-1)
      while(app < v[i]) AND (i > 0)
                                                     c4 · \sum (j=2 \rightarrow n)(t_w j)
      begin
          v[i+1] = v[i]
                                                      c5 · \Sigma(j=2 \rightarrow n)(t_wj)
                                                      c6 · \sum(j=2 → n)(t_wj)
       end
      v[i+1] = buff
                                                      c7 · (n-1)
   end
end
```

Formula per il calcolo delle operazioni

$$Tinss(n) = 4c(n-1) + 3c \Sigma(j=2 \rightarrow n) (t_w j)$$

 ${\bf Caso\ migliore:}\ (t_w j) {=}\ 0 \ \ \forall j \qquad \qquad [\ {\it Array\ gi\`{\bf a}\ ordinato}\]$

$$Tm(n) = 4c(n-1)+3c(0) = 4c(n-1) \approx n$$

$$\begin{split} Tp(n) &= 4c(n-1) + 3c(\Sigma(j=2 \to n)(j-1) \\ & \bullet &= 4c(n-1) + 3c[(n-1)(n)]/2 \approx n^2 \end{split}$$

$$\sum (j=2 \rightarrow n)(j-1) \approx \sum (j=1 \rightarrow n-1)j$$

Tempo medio: $t_w j = (j-1)/2$ [Si ferma circa a metà dei passi]