

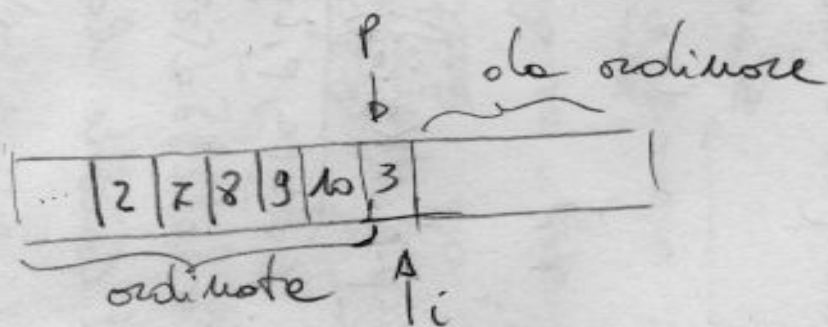
Insertion Sort

①

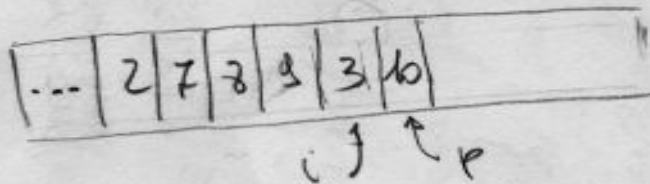
Isort

Idea

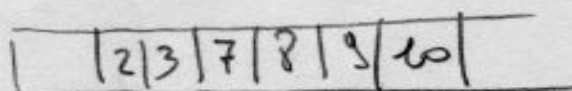
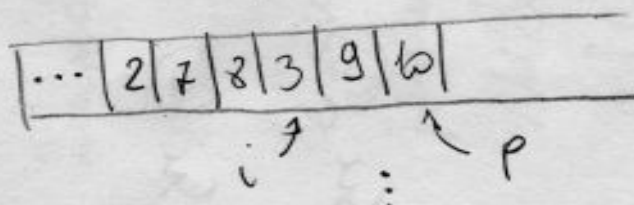
- 1) array diviso in due parti:
 -) $a[0] \dots a[p-1]$ ordinata
 -) $a[p] \dots a[n-1]$ non ordinata
- 2) assume che $i = p$ e sposta $a[i]$ verso sinistra finché $a[i] < a[i-1]$
- 3) si incrementa p ampliando la parte ordinata e ricomincia da 2).



$a[i] < a[i-1]$
↓ scambio



$a[i] < a[i-1]$
↓ scambio





Il algoritmo completa il suo lavoro esistendo
dentro delle porte ordinate per mezzo dell'ingestimento
di nuovi elementi con Shift Sort (a, p).

Insertion Sort (a)

```
p = 1  
while (p < a.length) {  
    ShiftSort(a, p)  
    p = p + 1;  
}
```

Prova

- ① con la formula $(n-1) + (n-2) + \dots + 2 + 1 = \frac{n(n-1)}{2}$ confronti e scambi
- ② se ogni elemento, nel momento in cui viene inserito nella parte ordinata non dista più di k posizioni, allora il costo è $M \cdot k$ confronti e scambi
- ③ è un limite nel senso che non ha niente di generale a.