range(0,5) 5 elementi

```python
def minimum_position(l, start):
    """
    Restituisce la posizione dell'elemento minimo nella sottolista l[start:].
    Si assume che start sia una posizione valida per l.
    """
    # La variabile minpos mantiene l'indice del valore più piccolo trovato
    # fino ad un dato momento. La inizializziamo con start.
    minpos = start
    # Facciamo variare l'indice i tra start+1 e la fine della lista, alla ricerca
    # del valore più piccolo
    for i in range(start + 1, len(l)):
        # Se il valore corrente è più piccolo di quello minimo trovato finora,
        # aggiorniamo minpos
        if l[i] < l[minpos]:
            minpos = i
    return minpos
```

$O(1)$

$O(1) * len(l) - start - 1$

$O(1) * (len(l) - start - 1)$

$O(1) * len(l) - start - 1$

$O(1)$

complessità della funzione minimum_position $O(len(l) - start - 1) = O(n)$

dove n è la lunghezza della "finestra di osservazione"

```python
def selection_sort(l):
    """Restituiscce gli stessi element di l ma ordinati."""
    # La variabile start indica l'inizio della finestra di l da ordinare. Faccio variare start da 0 a
    # len(l)-2. Ad ogni iterazione, trovo il minimo elemento nella finestrae, e lo metto in posizione
    # start. Quando start raggiunge len(l)-1 (quindi rimane un solo elemento da ordinare nella lista),
    # mi posso fermare perché una lista formata da un solo elemento è sicuramente ordinata.
    for start in range(len(l)-1):
        # Determino la posizione dell'elemento minimo nella sottolista di l che parte
        # alla posizione start. L'elemento che trovo andrà messo in posizione start.
        minpos = minimum_position(l, start)
        # Scambio l'elemento in posizione minpos con l'elemento in posizione start. In questo
        # modo l[start] assume il valore corretto. Lo scambio lo realizzo usando le tuple.
        l[start], l[minpos] = l[minpos], l[start]
```

$O(1) * ?$

la finestra di osservazione cambia tutte le volte, ma sempre più piccola della lunghezza di l

$O(len(l)) * ?$

$O(1) * ?$

Complessità selection sort

$O(1) * len(l) + O(len(l)) * len(l) + O(1) * len(l)$

$=$

$O(1) * n + O(n) * n + O(1) \cdot n$

dove n è la lunghezza della lista

$O(n) + O(n^2) + O(n) = O(n^2)$