Selection Sort

Wed, 16 Mar

L'algoritmo di **Selection Sort** è un algoritmo di **ordinamento**.

Istanza d'esempio

▼ *Selection Sort* applicato ad un *array* di esempio

- `72, 29, 27, 15, 78, 29`
- `15, 29, 27, 72, 78, 29`
- `15, 27, 29, 72, 78, 29`
- `15, 27, 29, 72, 78, 29`
- `15, 27, 29, 29, 78, 72`
- `15, 27, 29, 29, 72, 78`

Valutazione tempi di esecuzione

▼ **Pseudocodice** dell'algoritmo

```
void SelSort(V[])
begin
    for i = 1 to n-1                          c1 · (n-1)
    begin
        posmin = i                            c2 · (n-1)
        for j = i+1 to n                      c3 · (∑(i=0)(n-i))
        begin
            if(v[j] < v[posmin]               c4 · (∑(i=0 → n-i))
                posmin = j                    c5 · (t_i)
        end
        buff = v[i]                           c6 · (n-1)
        v[i] =v[posmin]                       c7 · (n-1)
        v[posmin] = buff                      c8 · (n-1)
    end
end
```

Formula per il calcolo delle **operazioni**

$$Tsels(n) = (n-1)(c1+c2+c6+c7+c8)+(\sum(i=0 \rightarrow n-i))(c3+c4)+c5(t_i)$$

▼ $$= (n-1)(c1+c2+c6+c7+c8)+[n(n-1)/2](c3+c4)+c5(t_i)$$

La sommatoria $\sum(i=0 \rightarrow n-i)$ si può scrivere come: [*ultimo_valore* + (*ultimo_valore* + *1*)] / 2

$$= 5c(n-1)+2c[n(n-1)/2]+c5(t_i)$$

Caso migliore: $t_i = 0$ [*Array già ordinato*]

$$Tm(n) = 5c(n-1) + 2c(n^2/2) - 2c(n/2) + 0 \approx n^2$$

Caso peggiore: $t_i = \sum(i=0)(n-1)i$ [*Array ordinato al contrario*]

$$Tp(n) = 5c(n-1) + 3c(n^2/2) - 2c(n/2) + 0 \approx n^2$$