

anno 2014-15  
Introduzione all'Algoritmica per i Licei

10 – Permutazioni

Elio Giovannetti  
Dipartimento di Informatica  
Università di Torino

1 marzo 2015



Quest'opera è distribuita con [Licenza Creative Commons](http://creativecommons.org/licenses/by-nc-sa/3.0/it/legalcode)  
[Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia.](http://creativecommons.org/licenses/by-nc-sa/3.0/it/legalcode)  
<http://creativecommons.org/licenses/by-nc-sa/3.0/it/legalcode>

# Monsieur Jourdain

- Nel “Borghese gentiluomo” di Molière il protagonista Monsieur Jourdain chiede al suo maestro di filosofia quali siano i diversi modi in cui può scrivere il biglietto galante: “**Bella Marchesa, i vostri begli occhi mi fanno morir d’amore**”
- Quattro secoli dopo, vogliamo aiutare Monsieur Jourdain a scrivere tutte le possibili varianti del suo biglietto.
- Più in generale, data una sequenza di elementi, vogliamo generare tutte le possibili permutazioni di tale sequenza,

## La lista di tutte le permutazioni di una lista

Vogliamo scrivere una funzione che, data una lista, dia come risultato una lista di liste: la lista di tutte le permutazioni della lista di partenza.

Esempio:

```
messaggio = ["i vostri occhi", "mi fanno", "morir", "d'amore"]
```

La lista `permutazioni(biglietto)` deve contenere le liste:

```
["i vostri occhi", "mi fanno", "morir", "d'amore"]
```

```
["mi fanno", "i vostri occhi", "morir", "d'amore"]
```

```
["morir", "i vostri occhi", "d'amore", "mi fanno"]
```

...

ecc.: quante sono tutte le possibili permutazioni?

# Una funzione ricorsiva

Per generare tutte le permutazioni di  $[a, b, c, d]$ :

- genero tutte le permutazioni il cui primo elemento è  $a$ :
  - genero tutte le permutazioni di  $[b, c, d]$
  - ad ognuna di esse metto in testa  $a$ ;
- genero tutte le permutazioni il cui primo elemento è  $b$ :
  - genero tutte le permutazioni di  $[a, c, d]$
  - ad ognuna di esse metto in testa  $b$ ;
- ecc.

Una lista vuota o di un solo elemento ha una sola permutazione, che è la lista stessa.

## La funzione

```
def permutazioni(lista):  
    result = []  
    n = len(lista)  
    if n == 1 or n == 0: return [lista]  
    else:  
        for i in range(n): # scegliamo il primo elemento  
            sceltoComePrimo = lista[i]  
            gliAltri = lista[:i] + lista[i+1:]  
            permutazioniAltri = permutazioni(gliAltri)  
            for perm in permutazioniAltri:  
                nuovaperm = [sceltoComePrimo] + perm  
                result.append(nuovaperm)  
    return result
```

Per visualizzare sullo schermo tutte le permutazioni in colonna:

```
messaggio = [  
    "i vostri occhi", "mi fanno", "morir", "d'amore"]  
for p in permutazioni(messaggio):  
    print(p)
```

## Il numero delle permutazioni.

Quante sono le possibili permutazioni di una lista di  $n$  elementi?

- Il primo elemento si può scegliere in  $n$  modi diversi;
- per ognuna di tali scelte il secondo si può scegliere in  $n-1$  modi, quindi abbiamo  $n(n-1)$  scelte possibili per i primi due elementi;
- per ognuna di tali  $n(n-1)$  scelte, il terzo si può scegliere in  $n-2$  modi, quindi si hanno  $n(n-1)(n-2)$  scelte possibili per i primi tre;
- e così via.

Per una lista di  $n$  elementi abbiamo quindi in tutto

$$n(n-1)(n-2) \dots 3 \cdot 2 \cdot 1$$

Tale numero si chiama **fattoriale di  $n$**  e si scrive:  $n!$

$$3! = 6, \quad 4! = 24, \quad 5! = 120, \quad 6! = 720, \quad 7! = 5040 \text{ ecc.}$$

Il fattoriale, al crescere di  $n$ , cresce più dell'esponenziale: è una funzione dalla crescita spaventosa!