

TUPLURI

Un *n-tuplu* e un șir de *n* elemente (x_1, x_2, \dots, x_n)

Proprietăți:

- elementele nu sunt neapărat distincte
- ordinea elementelor în tuplu contează

Cazuri particulare: *pereche* (a, b) , *triplet* (x, y, z) , etc.

etc.

15

Câte funcții există de la A la B ?

Dacă A și B sunt mulțimi finite există $|B|^{|A|}$ funcții de la A la B . (în fiecare element din B se poate mapa orice element din A)

Demonstrație: prin *inducție matematică* după $|A|$

Mulțimea funcțiilor $f : A \rightarrow B$ se notează uneori B^A

Mulțimea funcțiilor $f : A \rightarrow B$ se notează uneori B^A

Notăția ne amintește că numărul acestor funcții e $|B|^{|A|}$

24

Câte funcții injective există de la A la B ?

Dacă A și B sunt mulțimi finite și $f : A \rightarrow B$ injectivă
 $\Rightarrow |f(A)| = |A|$ (imaginea lui f va avea $|A|$ elemente).

Ordinea în care alegem elementele contază!

Ordinea în care alegem elementele *contează* !
(ordini diferite \Rightarrow funcții diferite)

... deci avem aranjamente de $|B|$ luate câte $|A|$

\Rightarrow există $A_{|B|}^{|A|} = \frac{|B|!}{(|B|-|A|)!}$ funcții injective

25

Câte funcții bijective există de la A la B ?

Dacă A și B sunt mulțimi finite și $f : A \rightarrow B$ bijectivă

$\Rightarrow |f(A)| = |A| = |B|$ (imaginea lui f va avea $|A|$

elemente)

$\Rightarrow |J(A)| = |A| = |B|$ (imaginea lui J va avea $|A|$ elemente).

Ordinea în care alegem elementele *contează* !
... deci avem permutări de $|A|$ elemente

\Rightarrow există $P/A| = |A|!$ Funcții bijective



Listă

Dacă dorim să definim o listă vom enumera între paranteze pătrate elementele acestui element

Dacă dorim să deținem o listă vom enumera între paranteze pătrate elementele acesteia, elementele fiind despărțite între ele prin virgulă:

`pare = [0, 2, 4, 6, 8]`

Dacă dorim să accesăm un element al listei procedăm la fel ca și în cazul șirurilor de caractere (Exemplu: dacă dorim să accesăm elementul 2 din lista *pare* vom scrie *pare[1]*, numerotarea făcându-se și în acest caz de la 0).

36

Tuplu

Un tuplu se definește folosind paranteze

Un tuplu se definește folosind paranteze rotunde:

```
tuplu_pare = (0, 2, 4, 6, 8)
```

37

Funcția poate fi returnată de alta
funcție în PYTHON

Python program to illustrate functions

FUNCTION DEFINITION

```
# Python program to illustrate functions  
# Functions can return another function
```

```
def create_adder(x):  
    def adder(y):  
        return x+y  
    return adder
```

```
add_15 = create_adder(15)  
print (add_15(10))
```

OUTPT:
25