

# Laboratorio di Python

## Esercizi sulle liste

16 aprile 2015

# Sommario

- 1 Correzione esercizi
- 2 Esercizi
- 3 Esercizi per casa



# Consegne

- 1 In quattro hanno consegnato gli esercizi alle 14:00 del 15/04.

# Copia del file

---

*#Scrivere una funzione che presi due file copi il primo nel secondo.*

```
def copia():  
    f1=open('primo.txt','r')  
    f2=open('secondo.txt','w')  
    righe1=f1.readlines()  
    f2.write(str(righe1))  
    f1.close()  
    f2.close()
```

---



# Estrazione del lotto

```
#Scrivere una funzione che simuli l'estrazione del lotto, tale  
#funzione deve scrivere su un file il risultato dell'estrazione.  
def estrazione():  
    import random  
    fl=open('estrazione.txt','w')  
    numeri=()  
    for i in range(1,91):  
#Creo l'insieme da cui estraggo i numeri  
        numeri=numeri+(i,)   
        estr=random.sample(numeri,5)  
#Estrazione  
        fl.write(str(estr))
```



# Esercizi sulle liste

- 1 Data una lista di coppie verificare se è il grafico di una funzione.
- 2 Data una lista di coppie che rappresenta una relazione, generare la relazione che è la sua chiusura transitiva.



# Esercizio 1

```
def seq(a):  
    return type(a)==list or type(a)==tuple  
  
def isrel(l):  
    if seq(l):  
        for i in range(0, len(l)):  
            if not seq(l[i]) or len(l[i])!=2:  
                return False  
        return True  
    else:  
        return False
```



# Esercizio 1

```
def funzione2(L):  
    if isrel(L):  
        S=[]  
        for c in L:  
            if c[0] not in S:  
                S.append(c[0])  
        return len(S)==len(L)  
    return False  
  
def funzione(L):    #più efficiente in media  
    if isrel(L):  
        S=[]  
        for c in L:  
            if c[0] in S:  
                return False  
            S.append(c[0])  
        return True  
    return False
```





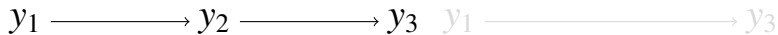
# Chiusura transitiva

Sia  $A$  una relazione su  $I^2$ ; si definisce chiusura transitiva di  $A$ , denotata con  $A^+$ , la relazione:

$A^+ = \{ \langle x, y \rangle \text{ t.c. Esistono } y_1, \dots, y_n \in A, \text{ con } n \geq 2, y_1 = x, y_n = y, \text{ t. c. } \langle y_i, y_{i+1} \rangle \in A, i = 1, \dots, n-1 \}$ .



# Chiusura Transitiva



$$A = \{ \langle y_1, y_2 \rangle, \langle y_2, y_3 \rangle \}$$

$$A^+ = \{ \langle y_1, y_2 \rangle, \langle y_2, y_3 \rangle, \langle y_1, y_3 \rangle \}$$



# Chiusura Transitiva

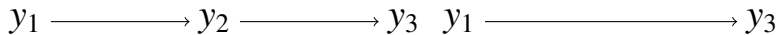


$$A = \{ \langle y_1, y_2 \rangle, \langle y_2, y_3 \rangle \}$$

$$A^+ = \{ \langle y_1, y_2 \rangle, \langle y_2, y_3 \rangle, \langle y_1, y_3 \rangle \}$$



# Chiusura Transitiva



$$A = \{ \langle y_1, y_2 \rangle, \langle y_2, y_3 \rangle \}$$

$$A^+ = \{ \langle y_1, y_2 \rangle, \langle y_2, y_3 \rangle, \langle y_1, y_3 \rangle \}$$



## Esercizio 2

Creiamo la soluzione a passi successivi:

- definiamo una nuova lista  $L$  con gli elementi della relazione iniziale  $R$ ,  $L$  rappresenta la variabile che conterrà la chiusura transitiva
- definisco una lista *nuove\_coppie* dove salvare le coppie da aggiungere ad  $L$
- per ogni coppia  $(x_1, x_2)$  e ogni coppia  $(x_2, x_3)$  di  $R$  inserisco la coppia  $(x_1, x_3)$  in *nuove\_coppie* se non è già presente in  $L$ , e in *nuove\_coppie*
- si aggiunge *nuove\_coppie* ad  $L$  e si controllerà se  $L$  genera altre possibili coppie appartenenti alla chiusura transitiva
- quando non si inseriscono più nuove coppie ad  $L$  allora  $L$  rispetta la proprietà di chiusura transitiva



## Esercizio 2

```
def transclos(R):  
    if not isrel(R):  
        return False  
    L=list(R)  
    devo_iterare=True #forzo inizio iterazione  
    while devo_iterare:  
        NuoveCoppie=[]  
        for c in L:  
            for x in L:  
                if c[1]==x[0] and (c[0],x[1]) not in L  
                    and (c[0],x[1]) not in NuoveCoppie:  
                    NuoveCoppie.append((c[0],x[1]))  
        L=L+NuoveCoppie  
        devo_iterare = (len(NuoveCoppie)!=0)  
    return L
```



# Esercizi

- 1 Scrivere una funzione che presa una lista e un valore  $i$  appartenente a tale lista restituisca due liste. La prima lista contiene i valori minori o uguali a  $i$  e la seconda lista i valori maggiori di  $i$  [non utilizzare il metodo sort].
- 2 Scrivere una funzione che presi come parametri due liste ordinate ne crei una sola anch'essa ordinata.
- 3 Scrivere una funzione che presa una lista controlli se al suo interno sono presenti valori uguali in posizioni successive, e in tal caso sostituire il valore uguale con il numero di occorrenze di valori uguali. (es.  $L=['a','b',1,1,'c'] \rightarrow O=['a','b',2,'c']$ )

Inviare gli esercizi svolti a: **labinfo.mat.unibo@gmail.com**



# Cosa abbiamo fatto?

- 1 Correzione esercizi
- 2 Esercizi
- 3 Esercizi per casa

