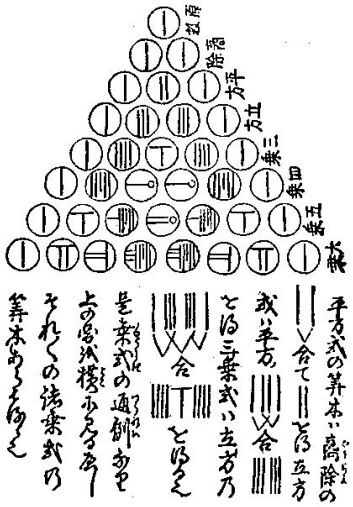
Matematica con Python

Mauro Saita

e-mail: maurosaita@tiscalinet.it

Versione provvisoria. Gennaio 2017.[1]

Lezione n. 4



Versione cinese del triangolo di Pascal.

Indice

[1] Nome File: python_lezione_04_2016.tex

1 Liste

Le liste sono sequenze (successioni finite) *modificabili* di elementi che posso essere di tipo diverso. L'i-esimo elemento della lista v è v[i], una lista costituita da un solo elemento si indica con [x], la lista vuota con [ ].

Definizione di una lista di nome v:

```
v = [1, 'Mauro', 2, 'Emiliano', 3, 'Michele']
```

Lettura (e scrittura) di un valore della lista v:

```
valore = v[3]
print(v[3])
#output: Emiliano
```

Visualizzazione di tutti gli elementi di una lista su una stessa riga:

```
>>> for i in range(6): print(v[i],end=' ')
```

oppure:

```
>>> for x in v:
        print(x,end=' ')
#output: 1 Mauro 2 Emiliano 3 Michele
```

Le liste possono essere annidate:

```
>>> a=[0,1,2,3,4]
>>> b=[5,6,7,8,9]
>>> c=[a,b]
>>> print(c)

#output: [[0,1,2,3,4],[5,6,7,8,9]]
```

oppure:

```
>>> for x in c:
    print(c)
```

```
#output: [0,1,2,3,4]
         [5,6,7,8,9]
```

Operazioni con le liste

Le principali operazioni che si possono applicare alle liste sono quelle elencate in tabella (v e w indicano due liste)

x in v	Vero, se x è un elemento della lista v	utilizzabile anche con stringhe e tuple
x not in v	Vero, se x non coincide con alcun elemento di v	utilizzabile anche con stringhe e tuple
v + w	Concatenazione di v e w	utilizzabile anche con stringhe e tuple
k * v	Concatenazione di k copie di v	utilizzabile anche con stringhe e tuple
v[i]	i-esimo elemento di v	utilizzabile anche con stringhe e tuple
v[-1]	Ultimo elemento di v	utilizzabile anche con stringhe e tuple
v[i:j]	Sequenza formata dagli elementi: v[i],v[i+1], . . . v[j-1]	utilizzabile anche con stringhe e tuple
v[:]	Copia di v.	utilizzabile anche con stringhe e tuple
len(v)	Lunghezza di v.	utilizzabile anche con stringhe e tuple
v.count(x)	Numero di volte che il valore x appare in v.	utilizzabile anche con stringhe e tuple
v.index(x)	Indice della prima posizione nella quale il valore x appare in v. Errore se x non è elemento di v	utilizzabile anche con stringhe e tple
min(v)	Valore minimo di v.	utilizzabile anche con stringhe e tuple
max(v)	Valore massimo di v.	utilizzabile anche con stringhe e tuple
sorted(v)	Lista che contiene gli elementi di v in ordine crescente. Si tratta di una lista anche se v è una stringa o una tupla.	utilizzabile anche con stringhe e tuple
reversed(v)	Inverte gli elementi di v.	utilizzabile anche con stringhe e tuple
v.append(x)	Aggiunge alla fine di v l'elemento x	utilizzabile solo con le liste
v.extend(x)	Aggiunge alla lista v la lista w	utilizzabile solo con le liste
v.insert(i,x)	Inserisce l'elemento x nella posizione i della lista v	utilizzabile solo con le liste
v.remove(x)	Elimina l'elemento x nella sua prima apparizione in v	utilizzabile solo con le liste
v.sort()	Ordina la lista in ordine alfabetico (crescente se gli elementi sono numeri). La lista viene modificata	utilizzabile solo con le liste

Esempi di operazioni che si possono fare su una lista (qui denominata v):

Aggiungere un elemento a una lista:

```
v=[0,1,2,3,4]
v.append(5)
print(v)

#output: [0, 1, 2, 3, 4, 5]
```

Eliminare un elemento da una lista

```
v=[0,1,2,3,4]
del v[3]
print(v)

#output: [0, 1, 2, 4]
```

Modificare un elemento di una lista

```
v=[0,1,2,3,4]
v[0] = 7
print(v)

#output: [7, 1, 2, 3, 4]
```

Lettura dell'ultimo elemento di una lista:

```
v=[0,1,2,3,4]
v[-1]

#output: 4
```

Appartenenza di un elemento ad un lista:

```
v=[0,1,2,3,4]
5 in v
#output: False
2 in v
#output: True
```

Concatenazione di liste:

```
v=[0,1,2,3,4]
w=[5,6,7,8,9]
v + w
#output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Ripetizione di liste:

```
v=[0,1,2,3,4]
v * 3
#output: [0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4]
```

2 Esercizi

Esercizio 2.1. *Cosrtruire liste diverse (a propria scelta) e provare, dalla shell di IDLE, tutti i comandi della precedente tabella.*

Esercizio 2.2. *Trovare da quante cifre sono formati i seguenti numeri:*

(a) 7^{13}

(b) 9^{11}

(c) 5^{19}

(I numeri sono espressi in base 10*).*

Esercizio 2.3. *Scrivere un programma che trova il numero più piccolo e quello più grande tra un elenco di numeri inseriti da tastiera.*

Esercizio 2.4. *Scrivere un programma che inverta una parola o un numero.*

3 Crivello di Eratostene

Il Crivello di Eratostene è un algoritmo che permette di trovare tutti i numeri primi compresi tra 1 e un prefissato intero positivo n. Per $n = 40$ l'algoritmo è il seguente:

- Si scrivono gli interi positivi da 1 fino a un numero 40

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

- Si elimina il numero 1. Si evidenzia il numero 2 (primo) e si eliminano i multipli di 2 fino a 40 (2 escluso)

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

- Si evidenzia il numero 3 (primo) e si eliminano i multipli di 3 fino a 40 (3 escluso)

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

- Il primo numero non ancora evidenziato o cancellato è 5: si evidenzia il numero 5 (primo) e si eliminano i suoi multipli fino a 40 (5 escluso)

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

- .....
- Il procedimento termina quando tutti i numeri dell'elenco sono stati evidenziati o cancellati

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

OSSERVAZIONE. Dopo aver evidenziato il numero 7 si scopre che:

(a) tutti i suoi multipli sono stati cancellati nei passaggi precedenti.

(b) i numeri non ancora cancellati, cioè $11, 13, 17, 19, 23, 29, 31, 37$, sono tutti primi ossia nessuno di questi verrà cancellato nei procedimenti successivi (perchè?).

L'algoritmo si può ottimizzare così

- Si scrivono gli interi positivi da 1 fino a n
- Si cancella il numero 1.
- .....
- Si evidenzia il più piccolo numero non ancora evidenziato o cancellato e si eliminano i suoi multipli fino a n.
- .....
- Il procedimento termina quando il più piccolo numero non ancora evidenziato è maggiore di $\sqrt{n}$.

Esercizio 3.1 (**Crivello di Eratostene**). *Scrivere un programma che, utilizzando il crivello di Eratostene, generi l'elenco dei numeri primi compresi tra* 1 *e* n*. Il programma deve:*

1. *creare una lista, denominata num, di interi, da* 1 *fino al numero* n*;*
2. *sostituire il primo elemento della lista (num[0]) con* 0*;*
3. *per ogni numero* x *da* 2 *a* $\sqrt{n}$ *controllare se l'elemento in posizione num[i] è multiplo di* x *(e diverso da* x*) porre num[i]=0;*
4. *eliminare tutti gli* 0 *dalla lista;*
5. *visualizzare la lista.*

4 Triangolo di Pascal

Si consideri il triangolo mostrato in figura

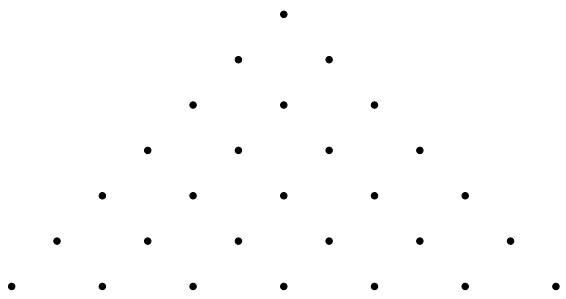


Figura 1

Cammini a zig zag.

Si chiama *cammino a zigzag* una spezzata che collega il vertice in alto a un punto qualsiasi del triangolo con queste proprietà

– il primo vertice della spezzata è il vertice in alto e l'ultimo coincide con il punto prescelto;

– ogni lato della spezzata collega un punto del triangolo con uno a scelta dei due punti che si trovano immediatamente al di sotto di esso.

In figura sono rappresentati due dei possibili cammini a zigzag che collegano il vertice in alto con quello denominato con $\diamond$.

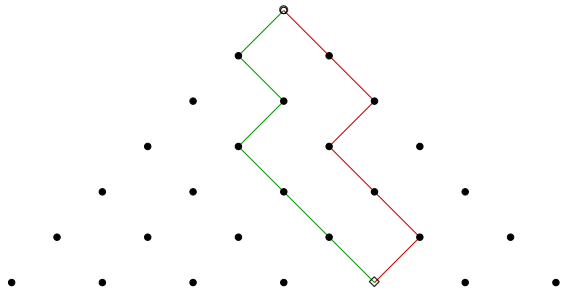


Figura 2: La spezzata rossa e quella verde sono due diversi "cammini a zigzag" che collegano il vertice in alto "$\circ$" con il punto del triangolo denotato con "$\diamond$".

Se si sostituisce ogni punto del triangolo della figura 1 con il numero di cammini a zigzag che lo collegano al vertice in alto si ottiene il ben noto "triangolo di Pascal":

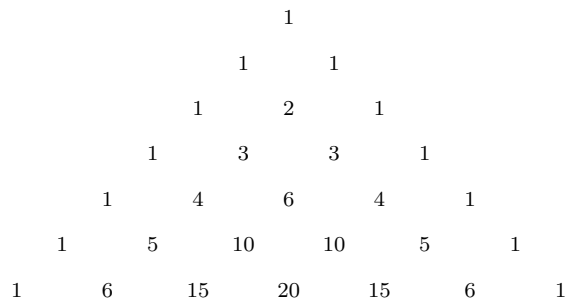


Figura 3: Il "triangolo di Pascal" è più noto in Italia con il nome di "triangolo di Tartaglia". Niccolò Tartaglia è il matematico italiano che per primo lo studiò e ne favorì la diffusione nella nostra penisola.

Esercizio 4.1 (**Triangolo di Pascal**)**.** *Utilizzando in modo opportuno le liste costruire le prime* 10 *righe del "triangolo di Pascal".*

Esercizio 4.2. *Del triangolo di Pascal calcolare*

1. *la somma degli elementi di ciascuna riga.*
2. *la somma con i segni alternati degli elementi di ciascuna riga.*

Quali congetture si possono formulare?

Esercizio 4.3. *Scrivere un programma che per ogni* $n \in \mathbb{N}$, $0 \leq n \leq 9$ *stampa lo sviluppo di*

$$(a+b)^n$$