Lezione 3 – Esercizi sui Dizionari

Informatica

22 Aprile 2015

Esercizio 1: Il negozio di libri

Scrivere un programma per gestire il magazzino di un negozio di libri. Il programma deve:

- tenere traccia di quante copie di ogni libro sono presenti in magazzino
- aggiungere una nuova copia di un libro in magazzino
- quando un libro viene venduto, aggiornare il magazzino
- stampare il contenuto del magazzino

```
def aggiungi_libro(lib,mag):
  if lib in mag:
    mag[lib]+=1
  else:
    mag[lib]=1

def vendi_libro(lib,mag):
  if lib in mag:
    if mag[lib]>1:
      mag[lib]-=1
    elif mag[lib]==1:
        del mag[lib]
    else:
      print("Libro in quantita' non valida")
  else:
    print("Libro non presente in magazzino")

def stampa_mag(mag):
  for lib in mag:
    print(lib,'*'*(mag[lib]),mag[lib])

print(" --- BOOKS ---")
mag=dict()
aggiungi_libro('Siddharta',mag)
aggiungi_libro('Siddharta',mag)
aggiungi_libro('Narciso e Boccadoro',mag)
aggiungi_libro('Io e Dio',mag)
vendi_libro('Io e Dio',mag)
stampa_mag(mag)
```

Esercizio 2: Rimuovere le lettere più frequenti

Scrivere una funzione che riceve in input una stringa e rimuove tutte le occorrenze dei caratteri a frequenza più alta.

```
def remove_most_frequent(a):
  b=dict()
  for e in a:
    b[e]=b.get(e,0)+1
  c=0
  for e in b:
    if b[e]>c:
      c=b[e]
  d=""
  for e in a:
    if b[e]<c:
      d=d+e
  return d

print(" --- REMOVE_MOST_FREQUENT --- ")
print(remove_most_frequent("aab"))
print(remove_most_frequent("babbaaccdabaa"))
```

Esercizio 3: L'alfabeto carbonaro

Durante i Moti del 1830-1831, gli aderenti alla Carboneria utilizzavano un *cifrario a sostituzione* per scambiarsi messaggi in codice.

Ogni lettera del messaggio da cifrare veniva sostituita da una lettera diversa, secondo una *tavola di accoppiamento*:

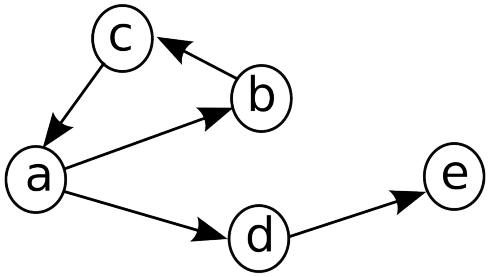
```
A|B|C|D|E|F|G|H|I|L|M|N|O|P|Q|R|S|T|U|V|Z
O|P|G|T|I|V|C|H|E|R|N|M|A|B|Q|L|Z|D|U|F|S
```

1. Utilizzare un dizionario per rappresentare la tavola di accoppiamento (*chiave*)
2. Scrivere una funzione per cifrare un testo
3. Scrivere una funzione per decifrare un testo

```
def cifra(chiave, testo):
    t = testo.lower()
    ris = ""
    for c in t:
        if c in chiave:
            ris = ris + chiave[c]
        else:
            ris = ris + c
    return ris

def inverti(chiave):
    ris = dict()
    for c in chiave:
        val = chiave[c]
        ris[val] = c
    return ris

def decifra(chiave, testo):
    chinv = inverti(chiave)
    return cifra(chinv, testo)

print(" --- CIFRARIO CARBONARO --- ")
chiave = {'a':'o', 'b':'p', 'c':'g', 'd':'t', 'e':'i', 'f':'v', 'g':'c',
```

```
          'h':'h', 'i':'e', 'l':'r', 'm':'n', 'n':'m', 'o':'a', 'p':'b',
          'q':'q', 'r':'l', 's':'z', 't':'d', 'u':'u', 'v':'f', 'z':'s'}
chiaro = "Preferisco la macchina alla moto."
print(chiaro)
msg = cifra(chiave, chiaro)
print(msg)
chiaro = decifra(chiave, msg)
print(chiaro)
```

Esercizio 4: visita di un grafo



1. rappresentare il grafo in figura con un dizionario
2. scrivere una funzione che visita il grafo a partire da un certo nodo
3. scrivere una funzione che controlla se un grafo è ciclico

```
def visita_aux(g, b, n):
    if n in g:
        if n not in b:
            b[n] = True
            print(n)
            for m in g[n]:
                visita_aux(g, b, m)

def visita(g, n):
    b = dict()
    visita_aux(g, b, n)

print(" --- GRAFO --- ")
grafo = {'a':('b','d'), 'b':('c',), 'c':('a',), 'd':('e',), 'e':()}
visita(grafo, 'a')

def ciclico_aux(g, b, n):
    if n in b:
        return b[n]
    b[n] = True
    for m in g[n]:
        if ciclico_aux(g, b, m):
            return True
    b[n] = False
    return False

def ciclico(g):
    b = dict()
```

```
    for n in g:
        if ciclico_aux(g,b,n):
            return True
    return False

print(ciclico(grafo))
grafo = {5:(11,), 11:(2,9,10), 7:(11,8), 8:(9,), 9:(), 3:(8,10), 10:(), 2:()}
print(ciclico(grafo))
```

Esercizio 5: Hapax

In linguistica, un *hapax* è una forma linguistica (parola o espressione), che compare una sola volta nell'ambito di un testo.

- scrivere una funzione che ritorni la lista di tutte le parole che compaiono una sola volta in un testo

```
spazi = {' ':1, ',':1, ';':1, '.':1, "'":1, '"':1, '!':1, '?':1}

def hapax(s):
    freq = dict()
    parola = ""
    for c in s:
        if c not in spazi:
            parola += c
        else:
            if parola != "":
                freq[parola] = freq.get(parola,0) + 1
                parola = ""
    if parola != "":
        freq[parola] = freq.get(parola,0) + 1
    l = []
    for p in freq:
        if freq[p] == 1:
            l.append(p)
    return l

print(" --- HAPAX --- ")
print(hapax("La mattina seguente Don Rodrigo si destò Don Rodrigo"))
```