Table of Contents

RECAP:
Rotazione ricorsiva di immagini
  Suddivisione e rotazione ricorsiva NxN
Esercizi d'esame

Fondamenti di Programmazione

Andrea Sterbini

lezione 20 - 15 dicembre 2022

RECAP:

- albero di gioco del **Tris**/**Filetto**
  - rappresentazione
  - generazione
  - configurazioni equivalenti
  - strategie
- rappresentazione di **espressioni algebriche**
  - calcolo
  - semplificazione
  - parsing

Rotazione ricorsiva di immagini

`| 1 |` **resta com'è**

```
1 | 2
--|--
3 | 4
```

diventa

```
2 | 4
--|--
1 | 3
```

```
 0 1 | 2 3
 4 5 | 6 7
 ----|----
 8 9 | A B
 C D | E F
```

diventa

```
3 7 | B F
2 6 | A E
----|----
1 5 | 9 D
```

Suddivisione e rotazione ricorsiva NxN

- se N==1: la matrice resta così (**caso base**)
- se N>1:
  - divido in 4 sottomatrici (**riduzione**)
  - le ruoto di 90° (**chiamata ricorsiva**)
  - le scambio
  - le fondo in una matrice più grande (**composizione**)
- a forza di dividere per 2 arrivo sempre a 1 (**convergenza**)

In [1]:
```python
# A B
# C D

def dividiP2(matrice):
    "divido la matrice nei suoi 4 quadranti"
    # NOTA: la matrice deve avere dimensione potenza di 2
    L = len(matrice)
    assert L>1 and L%2==0 # FIXME: dovrei controllare 2^n==L
    metà = L//2
    fascia_superiore = matrice[:metà]
    fascia_inferiore = matrice[metà:]
    A = [ riga[:metà] for riga in fascia_superiore ]
    B = [ riga[metà:] for riga in fascia_superiore ]
    C = [ riga[:metà] for riga in fascia_inferiore ]
    D = [ riga[metà:] for riga in fascia_inferiore ]
    return A, B, C, D
```

In [6]:
```python
# vediamo se funziona
M = [[1,2,3,4],
     [5,6,7,8],
     [9,10,11,12],
     [13,14,15,16]]
N = [[1,2],[3,4]]
##print(*dividiP2(M),sep='\n\n')
```

```
[[1, 2], [5, 6]]

[[3, 4], [7, 8]]

[[9, 10], [13, 14]]

[[11, 12], [15, 16]]
```

In [7]:
```python
# A B
# C D
def fondiP2(A, B, C, D):
    "fondo 4 matrici in una sola"
    AB = [ rigaA+rigaB for rigaA,rigaB in zip(A,B)  ]
    CD = [ rigaC+rigaD for rigaC,rigaD in zip(C,D)  ]
    return AB + CD
```

In [8]:
```
# test
A = [[1,2],[5,6]]
B = [[3,4],[7,8]]
C = [[9,10],[13,14]]
D = [[11,12],[15,16]]
print(*fondiP2(A,B,C,D), sep='\n')
```

```
[1, 2, 3, 4]
[5, 6, 7, 8]
[9, 10, 11, 12]
[13, 14, 15, 16]
```

In [9]:
```
# A B
# C D

# B D
# A C

def ruotaP2(M):
    "ruoto una matrice che ha lato 2^L"
    # se N==1 la ritorno tal quale
    if len(M) == 1:
        return M
    # la divido in 4
    A, B, C, D = dividiP2(M)
    # le ruoto tutte e 4
    Aruotata = ruotaP2(A)
    Bruotata = ruotaP2(B)
    Cruotata = ruotaP2(C)
    Druotata = ruotaP2(D)
    # le scambio e le fondo
    return fondiP2(Bruotata, Druotata, Aruotata, Cruotata)
```

In [10]:
```
M = [['1', '2', '3', '4'],
     ['5', '6', '7', '8'],
     ['9', 'A', 'B', 'C'],
     ['D', 'E', 'F', 'G']]
print(*ruotaP2(M), sep='\n')
```

```
['4', '8', 'C', 'G']
['3', '7', 'B', 'F']
['2', '6', 'A', 'E']
['1', '5', '9', 'D']
```

In [26]:

```python
def ruota(M):
    "generico ruota applicabile a matrici != da potenza di 2"
    # allargo la matrice alla prossima potenza di 2 nelle due direzioni
    W = len(M[0])
    H = len(M)
    i = 0
    while 2**i < W:
        i +=1
    larghezza = 2**i
    i = 0
    while 2**i < H:
        i +=1
    altezza = 2**i
    for riga in M:
        riga += [(0,0,0)] * (larghezza-W)
    M += [ [(0,0,0)] * larghezza for i in range(altezza-H) ]
    # ruoto la matrice
    ruotata = ruotaP2(M)
    # elimino le righe/colonne aggiunte
    ruotata = ruotata[larghezza-W:]
    return [ riga[:H] for riga in ruotata  ]
```

In [27]:

```python

#  3 7 B F
#  2 6 A E
#  1 5 9 D
#  0 4 8 C
M = [ ['0', '1', '2', ],
      ['4', '5', '6', ],
      ['8', '9', 'A', ]]

print(*ruota(M), sep='\n')
```

```
['2', '6', 'A']
['1', '5', '9']
['0', '4', '8']
```

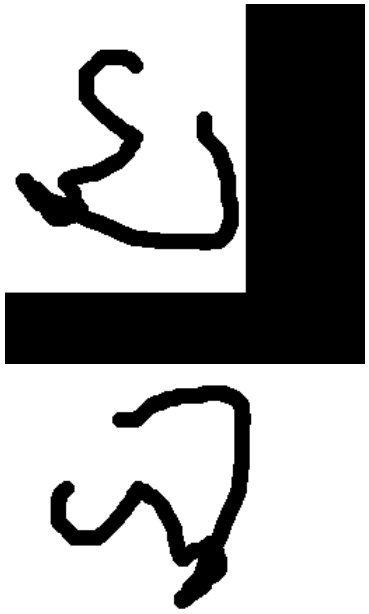
In [28]:

```
import images

img = images.load('scarabocchio.png')

ruotata = ruota(img)

images.save(ruotata, 'scarabocchio-90°.png')
images.visd(img), images.visd(ruotata)
```



Out[28]: (None, None)

In [29]:

```
# %%%

lenna = images.load('Lenna.png')
ruotata = ruota(lenna)
images.save(ruotata, 'Lenna-ruotata.png')

images.visd(lenna), images.visd(ruotata)
```

Out[29]: (None, None)

Esercizi d'esame