

 POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Quarta esercitazione

Riccardo Cattaneo



POLITECNICO
DI MILANO

11/11/2013



Terza esercitazione: agenda

- (15') Es1: somma virgola fissa e mobile
- (05') Es2: somma binaria algebrica CPL2
- (20') Es3: matrice di incidenza
- (20') Es4: ristorante
- (20') Es5: schermo led
- (10') Pausa
- (40') Es6: Conway's Game of Life
- Esercizio bonus - email

Es1: Somme in virgola mobile

$$37,53125 + 22,3125$$

37	1	0,53125	1	22	0	0,3125	0
18	0	0,06250	0	11	1	0,6250	1
9	1	0,12500	0	5	1	0,2500	0
4	0	0,25000	0	2	0	0,5000	1
2	0	0,50000	1	1	1	0	END
1	1	0	FINE	0	FINE		
0	1						

(Fix) 100101.10001

(Fix) 10110.0101

Es1: Somme in virgola mobile

$$(-1)^S \times 1.m \times 2^{(e-127)}$$

100101.10001

↓ NORM.

$$1.0010110001 \times 2^5$$

5 bit

$$S = 0 \quad (1 \text{ bit})$$

$$(positive) m = 0010110001 \dots 0 \quad (23 \text{ bit})$$

$$e = 5 + 127 = 132 \Rightarrow 10000100 \quad (8 \text{ bit})$$

(132₁₀)

$$e \cdot 2^N$$

↓

$$e \cdot 2^N$$

$$\longrightarrow (e + b) \cdot 2^N$$

10110.0101

↓ NORM

$$1.01100101 \times 2^4$$

4 bit

$$S = 0 \quad (1 \text{ bit})$$

$$(positive) m = 01100101 \dots 0 \quad (23 \text{ bit})$$

$$e = 4 + 127 = 131 \Rightarrow 1000011 \quad (8 \text{ bit})$$

(131₁₀)

$$b \cdot 2^{N-1}$$

30×10^0
 30×10^1
 3×10^2
 3.3×10^3
 0.03×10^4

↓

$$0.6 \times 2^N$$

Es1: Somme in virgola mobile

1.0010110001

1.01100101



0.101100101 (DENORMAL!)



1.0010110001 +

0.1011001010 =

1.1101111011 (x 2¹³²) EXTENDED, OK!

1110.11.11011

↓

32+16+8+2+1

59₁₀

0,5 +
0,25 +
0,0625 + 0,03125 = 0,9375 = 0,94375

59,94375 c.v.d.

Terza esercitazione: agenda

- (15') Es1: somma virgola fissa e mobile
- (05') Es2: somma binaria algebrica CPL2
- (20') Es3: matrice di incidenza
- (20') Es4: ristorante
- (20') Es5: schermo led
- (10') Pausa
- (40') Es6: Conway's Game of Life
- Esercizio bonus - email

Es2. Somma binaria algebrica CPL2

$$\begin{array}{ccc} \text{4 bit} & \text{4 bit} & \text{5 bit} \\ -12 & -15 & = -27 \end{array}$$

$$\begin{array}{rcll} 12 & 1100 & 01100 & 10011 \quad 10100 + \\ 15 & 1111 & 01111 & 10000 \quad 10001 = \end{array}$$

$$\begin{array}{r} 100101 \\ \hline 100100 \\ \hline 111011 \end{array}$$

Carry
CPL1

168 21

27

(SICCOME È PASSATO
PER CPL2, NEGATIVO)

Terza esercitazione: agenda

- (15') Es1: somma virgola fissa e mobile
- (05') Es2: somma binaria algebrica CPL2
- (20') Es3: matrice di incidenza
- (20') Es4: ristorante
- (20') Es5: schermo led
- (10') Pausa
- (40') Es6: Conway's Game of Life
- Esercizio bonus - email

Es3. Matrici di incidenza

Si chieda all'utente di inserire da tastiera *al più* N coppie di interi nella forma $(n1, n2)$. Se l'utente inserisce il valore $(-1, -1)$ oppure viene inserito il valore $N-1$ -esimo, il programma stampa la matrice di incidenza, definita come la matrice binaria $N \times N$ nulla, ad eccezione delle celle identificate dalle posizioni $(n1, n2)$ inserite dall'utente, dove il valore è 1.

(utile per rappresentare in forma matriciale relazioni e funzioni)

Terza esercitazione: agenda

- (15') Es1: somma virgola fissa e mobile
- (05') Es2: somma binaria algebrica CPL2
- (20') Es3: matrice di incidenza
- (20') Es4: ristorante
- (20') Es5: schermo led
- (10') Pausa
- (40') Es6: Conway's Game of Life
- Esercizio bonus - email

Es4. Ristorante

Si ipotizzi di dover gestire un sistema informatico per un ristorante. Si caratterizzino le strutture dati che descrivono un menu, ossia: **menu** (che ha al massimo 15 piatti per categoria, tre categorie: primi, secondi, dolci), **piatto** (costituito da al massimo 3 ingredienti di tipo stringa ed una ricetta di tipo stringa, molto lunga), **prezzo** (netto ed IVA% di tipo float).

Si ipotizzi quindi che qualcuno vi abbia fornito un menù; si scriva la funzione che cerca il primo di pasta più economico.

Terza esercitazione: agenda

- (15') Es1: somma virgola fissa e mobile
- (05') Es2: somma binaria algebrica CPL2
- (20') Es3: matrice di incidenza
- (20') Es4: ristorante
- (20') Es5: schermo led
- (10') Pausa
- (40') Es6: Conway's Game of Life

Es5. Schermo LED

Si ipotizzi di dover gestire uno schermo LED. Ogni pixel dello schermo, una matrice da 1920x1080 pixel, contiene una tripla di valori, ossia R_{ed} , G_{reen} e B_{lue} da rappresentare come struct.

Si scriva quindi una funzione che ritorna il valore di illuminazione *medio* di *solo un* canale (tra R, G e B) di una colonna di pixel. Canale e colonna sono a scelta dell'utente.

Es6. Conway's Game of Life

Avete una matrice $M \times N$; ogni cella può essere *viva* o *morta*. La matrice “evolve” dal tempo t al tempo $t+1$ secondo queste regole:

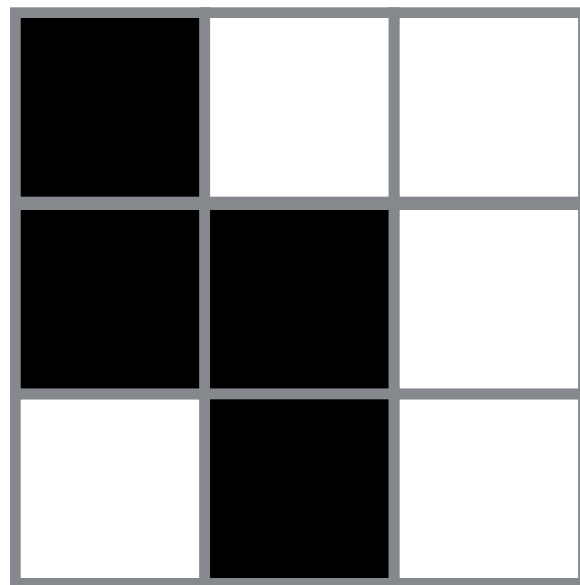
- 1) una cella morta con esattamente 3 vicini vivi *nasce* diventando viva
- 2) una cella viva con 2 o 3 vicini vivi *sopravvive*
- 3) una cella viva con 1 *solo* vicino o con *almeno* 4 vicini vivi muore.

Scrivete un pezzo di codice che evolve una matrice $M \times N$ per K iterazioni, quindi stampa la matrice (1 se viva, 0 se morta)

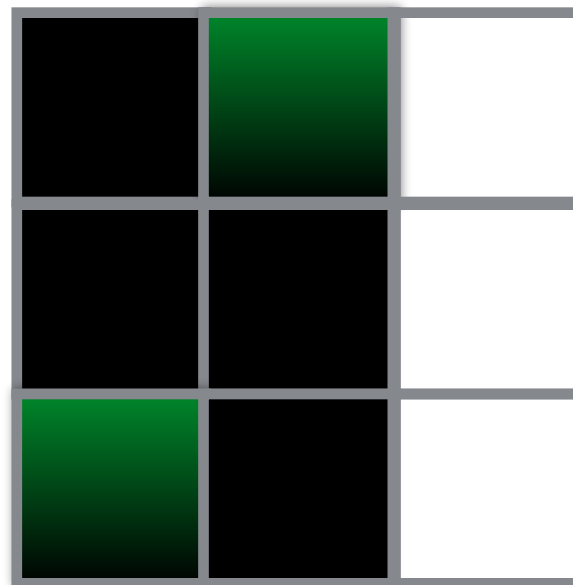
Si ipotizzi di chiedere all'utente di inserire 4 celle vive di partenza, in un formato opportuno.

Es6. Conway's Game of Life

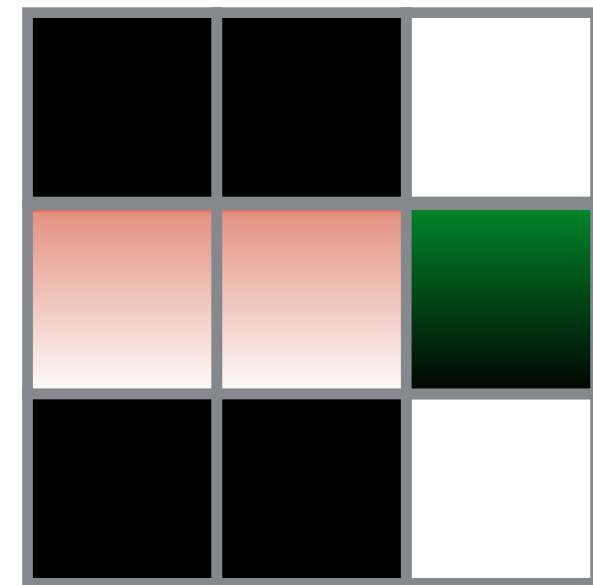
Inizio



Regola 1



Regola 1, 2 e 3

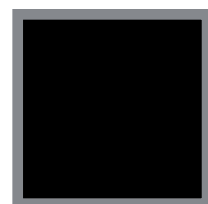


0



K

1



Viva

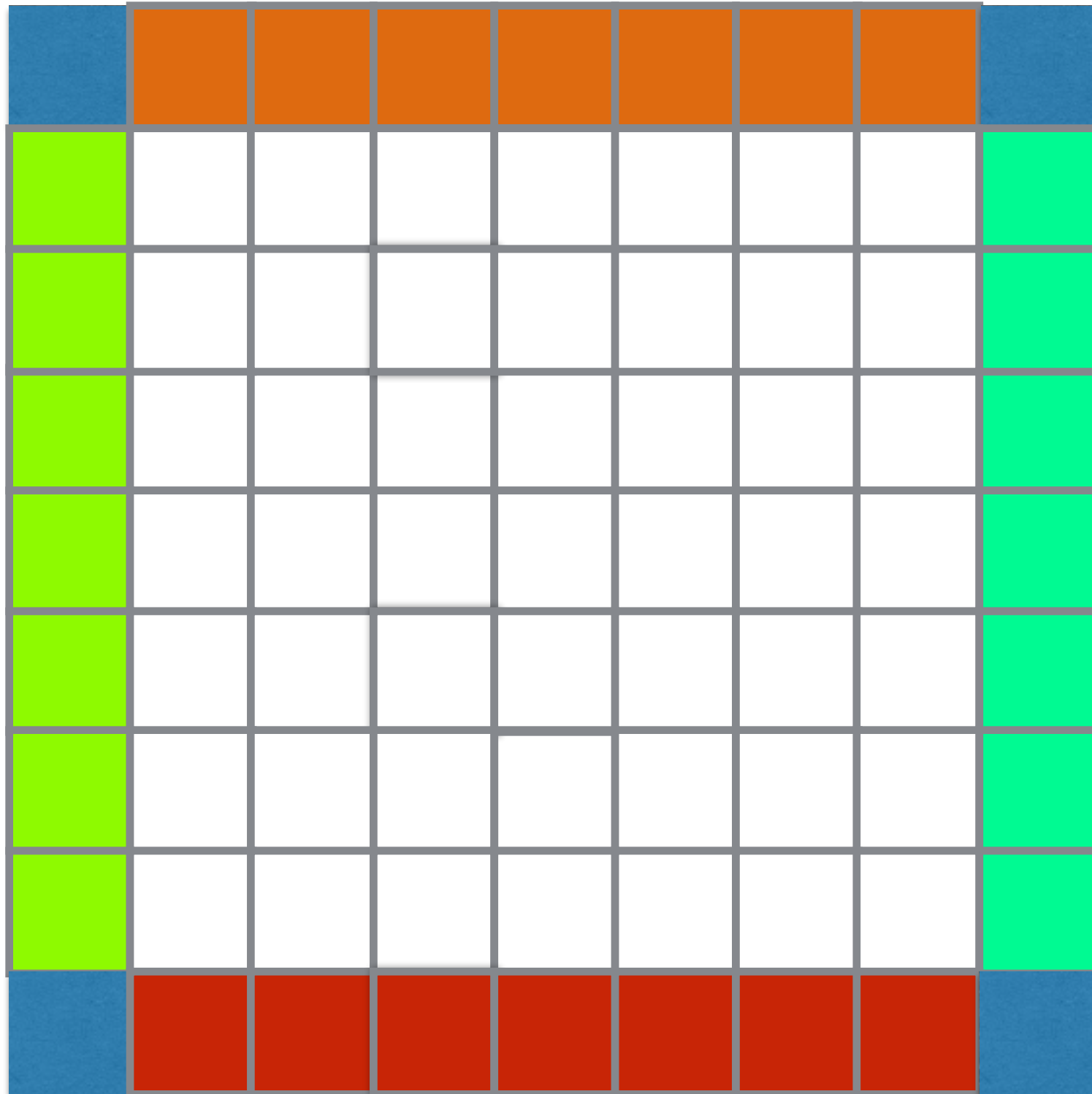


Morta in questa iterazione



Nata in questa iterazione

Es6. Conway's Game of Life



Attenzione agli “effetti di bordo”. Richiedono un trattamento particolare nella definizione delle condizioni perché hanno *sempre* meno di 8 vicini!

Bonus. Filtri di convoluzione 2D

Realizzare un codice che calcoli il risultato dell'applicazione di un filtro di convoluzione 2D su un segnale bidimensionale di ingresso indicato come $s[M][N]$. Si supponga $M, N > 3$, si trascurino gli effetti di bordo (imporre il segnale al bordo pari al valore originale) e si consideri un filtro $h(i,j)$ 3x3 come in figura. L'equazione aggiorna la "cella" a tempo $t+1$ analogamente a quanto visto con Conway's Game of Life, e riassume la/e regola/e di update.

$$F(x, y) = f(x, y) * g(x, y) = \sum_i \sum_j f(x + i, y + j) h(i, j)$$

1 / 16

1	2	1
2	4	2
1	2	1