Titolo unità didattica: Formule ricorrenti [12]

Titolo modulo : Algoritmi per le formule ricorrenti [01-T]

Approccio incrementale e formule ricorrenti

Argomenti trattati:

- ✓ ordine di una formula ricorrente
- ✓ condizioni iniziali
- ✓ formule ricorrenti lineari
- ✓ formule ricorrenti e approccio incrementale
- ✓ le formule di Malthus e logistica

Prerequisiti richiesti: P1-06-01-T

formule ricorrenti

una **formula** **ricorrente** (o **relazione di** **ricorrenza**) è una equazione che definisce **ricorsivamente** una **successione**

Esempio: la successione $1, 3, 7, 15, 31, \ldots$

è definita ricorsivamente dalla formula ricorrente

$$\boxed{y_k = 2y_{k-1} + 1}$$

y_k k-simo termine della successione

y_{k-1} $(k-1)$-simo termine della successione

y_k è espresso come funzione del termine precedente y_{k-1}

formule ricorrenti

una **formula** **ricorrente** (o **relazione di** **ricorrenza**) è una equazione che definisce **ricorsivamente** una **successione**

Esempio: la successione $1, 3, 7, 15, 31, \ldots$

è definita ricorsivamente dalla formula ricorrente

$$\boxed{y_k = 2y_{k-1} + 1}$$

y_0 termine iniziale della successione $y_0 = 1$

fissato il termine iniziale y_0 si ha una successione $\{y_k\}_{k=0}^{\infty}$

condizione iniziale

formule ricorrenti

Esempio

$$y_k = 2y_{k-1} + 1$$

$$y_0 = 1$$

$k = 1$ $\quad y_1 = 2y_0 + 1$ $\quad y_1 = 2 \cdot 1 + 1$ $\quad y_1 = 3$

$k = 2$ $\quad y_2 = 2y_1 + 1$ $\quad y_2 = 2 \cdot 3 + 1$ $\quad y_2 = 7$

$k = 3$ $\quad y_3 = 2y_2 + 1$ $\quad y_3 = 2 \cdot 7 + 1$ $\quad y_3 = 15$

....

formule ricorrenti

Esempio

$$y_k = 2y_{k-1} + 1$$

$$y_0 = 10$$

$k = 1$ $\quad y_1 = 2y_0 + 1 \quad y_1 = 2 \cdot 10 + 1 \quad y_1 = 21$

$k = 2$ $\quad y_2 = 2y_1 + 1 \quad y_2 = 2 \cdot 21 + 1 \quad y_2 = 43$

$k = 3$ $\quad y_3 = 2y_2 + 1 \quad y_3 = 2 \cdot 43 + 1 \quad y_3 = 87$

....

formula ricorrente del **primo ordine**

y_k dipende **solo** dal termine precedente y_{k-1}

$$y_k = a y_{k-1} + b$$

lineare, a coefficienti **costanti**

$$y_k = a_k y_{k-1} + b_k$$

lineare, a coefficienti **non costanti**

$$y_k = a y_{k-1}^2$$

nonlineare, a coefficienti **costanti**

$\Downarrow$

una **sola** condizione iniziale da fissare y_0

formula ricorrente del **secondo ordine**

y_k dipende **solo** dai termini y_{k-1}, y_{k-2}

$$y_k = a y_{k-1} + b y_{k-2} + c$$

lineare, a coefficienti **costanti**

$$y_k = a_k y_{k-1} + b_k y_{k-2} + c_k$$

lineare, a coefficienti **non costanti**

$$y_k = a y_{k-1} y_{k-2}$$

nonlineare, a coefficienti **costanti**

$\Downarrow$

due condizioni iniziali da fissare y_0 y_1

problema: visualizzare i primi $n+1$ termini generati dalla formula ricorrente del I ordine, lineare a coefficienti costanti

$$y_k = a y_{k-1} + b$$

dati di input: n, a, b, y_0 (variabili **n**,**a**, **b**, **y_zero**)

dato di output: nessuno

costrutto ripetitivo: **for**

operazione ripetuta (al generico passo **k**)**:**

applicazione della formula ricorrente per determinare valore del **k**-simo termine della successione (variabile **y**)

non deve essere usato un array

```
void ricorrente_1lc(int n, float a, float
b, float y_zero) {
 float y;
 int k;
 y = y_zero ;
 printf (y) ;
 for (k=1; k<=n; k++) {
      y = a*y+b ;
      printf (y) ;
  }
}
```

il nuovo valore di y (cioè y_k)
è il vecchio valore di y
(cioè y_{k-1})
moltiplicato per a e sommato a b

non deve essere usato un array

problema: calcolare il valore dell'n-simo termine della successione generata dalla formula ricorrente

$$y_k = a y_{k-1} + b$$

```
float form_ric_1lc(int n, float a, float b,
      float y_zero) {
 float y;
 int k;
 y = y_zero ;
 for (k=1; k<=n; k++) {
      y = a*y+b ;
 }
 return y ;
}
```

problema: visualizzare il valore dei primi n termini e del valore assoluto della differenza di ogni termine con il precedente di una formula ricorrente

Esempio:

$$y_k = 2y_{k-1} + 1 \qquad y_0 = 10$$

$$10,\ 21,\ 43,\ 87, \ldots\ldots$$

$$11,\ 22,\ 44, \ldots\ldots$$

problema: visualizzare il valore dei primi *n* termini e del valore assoluto della differenza di ogni termine con il precedente di una formula ricorrente

```
void ricorrente_1lc(int n, float a, float b, float y_zero) {
 float y_att,y_prec;
 int k;
 y_prec = y_zero ;
 printf (y_prec) ;
 for (k=1; k<=n; k++) {
      y_att = a*y_prec+b ;
      printf (y_att,abs(y_att-y_prec)) ;
      y_prec = y_att ;
  }
}
```

aggiornamento della “memoria”

problema: possiamo esprimere in forma chiusa il generico termine y_k per qualunque valore di k **?**

$\boxed{y_k = a y_{k-1} + \cancel{}}$

$$y_k = a y_{k-1}$$

y_0 fissato (>0)

$$y_1 = a y_0$$

$$y_2 = a y_1 = a a y_0 = a^2 y_0$$

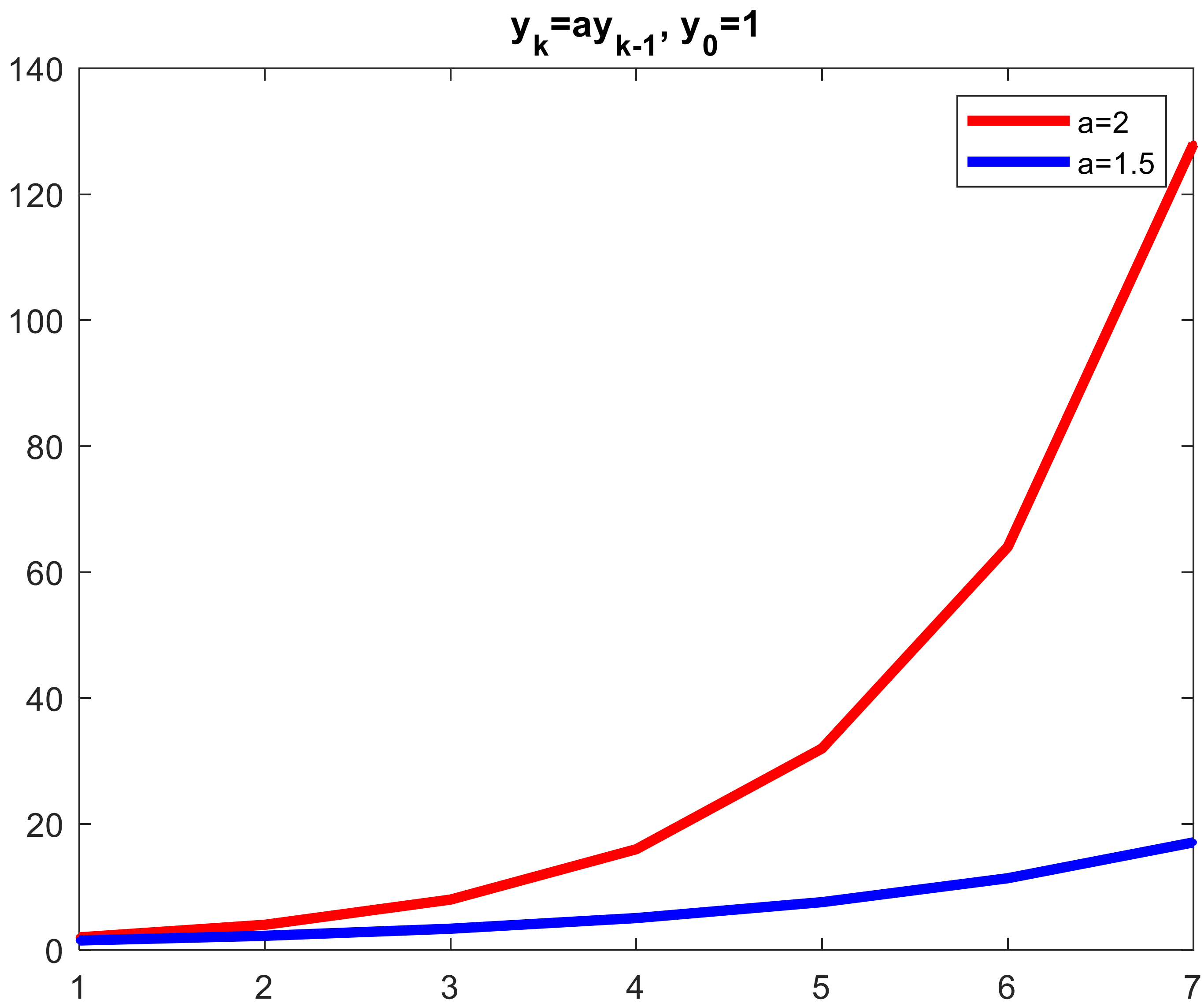
$$y_3 = a y_2 = a a a y_0 = a^3 y_0$$

$$y_k = a y_{k-1} = a a^{k-1} y_0 = a^k y_0$$

$$\boxed{y_k = a^k y_0}$$

$a > 1$ crescita esponenziale

$0 < a < 1$ decrescita esponenziale verso 0

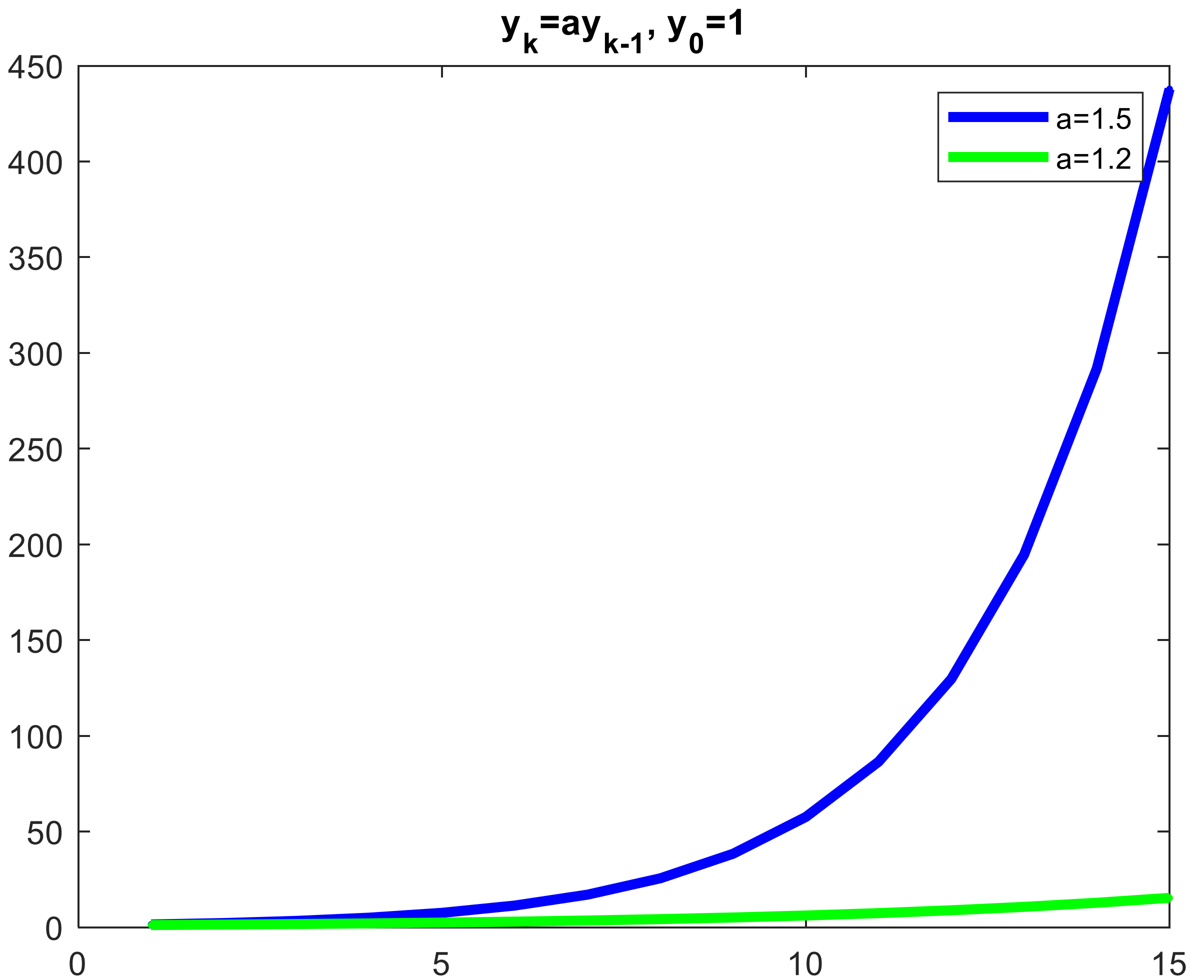

$y_k = ay_{k-1},\ y_0 = 1$
a=2
a=1.5
0
20
40
60
80
100
120
140
1
2
3
4
5
6
7

tempo di raddoppio

$$k_{radd}: y_{k_{radd}} = 2y_0$$

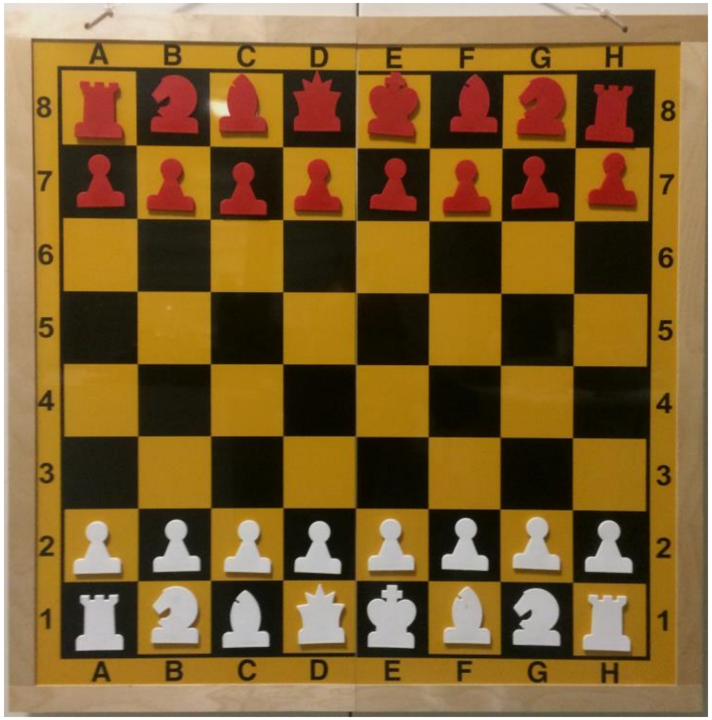
$$a^{k_{radd}} y_0 = 2y_0$$

$$k_{radd} = \left(\frac{1}{\log_2 a}\right)$$



$$k_{radd} = \left(\frac{1}{\log_2 a}\right) = \ 1.7\ ,\ 3.8$$

l’aneddoto del faraone e del suo salvatore



«solo un chicco di riso! ….

che si raddoppia a ogni casella»

64 caselle

2^{63} chicchi di riso

1 chicco di riso pesa 1/45 g

2^{63} chicchi di riso pesano 2.05 *10^{17} g = 2.05*10^{11} tonnellate

205 miliardi di tonnellate

produzione mondiale di riso anno 2022:
2.9 miliardi di tonnellate

problema: visualizzare i primi $n+1$ termini generati dalla formula ricorrente del II ordine, lineare a coefficienti costanti

$$\boxed{y_k = ay_{k-1} + by_{k-2} + c}$$

dati di input: n, a, b, c, y_0, y_1 (variabili **n**,**a**,**b**,**c**, **y_zero**, **y_uno**)

dato di output: nessuno

costrutto ripetitivo: **for**

operazione ripetuta (al generico passo **k**)**:**

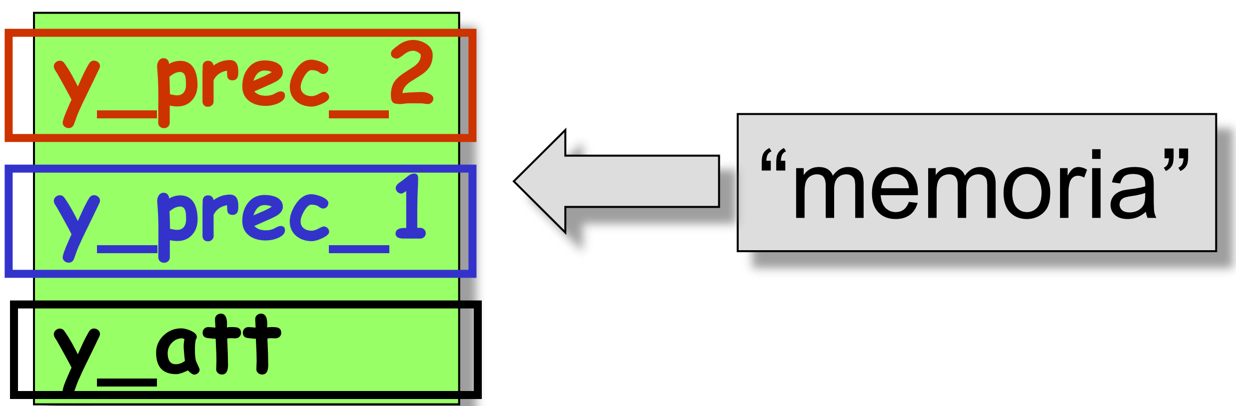
applicazione della formula ricorrente per determinare valore del **k**-simo termine della successione (variabile **y**)

non deve essere usato un array

problema: visualizzare i primi $n+1$ termini generati dalla formula ricorrente del II ordine, lineare a coefficienti costanti

$$\boxed{y_k = a y_{k-1} + b y_{k-2} + c}$$

$$y_0, y_1, y_2, y_3, \ldots, \boxed{y_{k-2}}, \boxed{y_{k-1}}, \boxed{y_k}$$



problema: visualizzare i primi $n+1$ termini generati dalla formula ricorrente del II ordine, lineare a coefficienti costanti

$$y_k = a y_{k-1} + b y_{k-2} + c$$

$$\boxed{y_0, y_1, y_2}, y_3, \ldots, y_{k-2}, y_{k-1}, y_k$$

y_prec_2
y_prec_1
y_att

primo passo

problema: visualizzare i primi $n+1$ termini generati dalla formula ricorrente del II ordine, lineare a coefficienti costanti

$$\boxed{y_k = a y_{k-1} + b y_{k-2} + c}$$

$$y_0, \boxed{y_1, y_2, y_3}, \dots, y_{k-2}, y_{k-1}, y_k$$

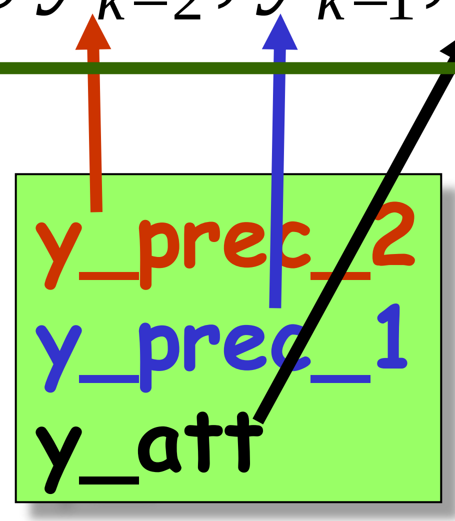
y_prec_2
y_prec_1
y_att

secondo passo

problema: visualizzare i primi $n+1$ termini generati dalla formula ricorrente del II ordine, lineare a coefficienti costanti

$$\boxed{y_k = a y_{k-1} + b y_{k-2} + c}$$

$$y_0, y_1, y_2, y_3, \dots, \boxed{y_{k-2}, y_{k-1}, y_k}$$



k-simo passo

problema: visualizzare i primi $n+1$ termini generati dalla formula ricorrente del II ordine, lineare a coefficienti costanti

```
void form_ric_2lc(int n, float a, float b, float c, float
                    y_zero, float y_uno) {
 float y_att, y_prec_1,y_prec_2;
 int k;
 y_prec_2 = y_zero, y_prec_1 = y_uno ;
 printf (y_prec_1,y_prec_2) ;
 for (k=2; k <= n; k++) {
    y_att = a*y_prec_1+b*y_prec_2+c ;
    printf (y_att,abs(y_att-y_prec_1)) ;
    y_prec_2 = y_prec_1 ;
    y_prec_1 = y_att ;
  }
}
```

la **memoria** necessaria all'algoritmo è costituita dalle sole due variabili

y_prec_1 e **y_prec_2**

aggiornate alla fine di ogni passo del ciclo

```
y_att = calcolo termine corrente
y_prec_2 = y_prec_1
y_prec_1 = y_att
```

struttura dell'algoritmo per la valutazione di formule ricorrenti (tecnica **iterativa**)

dipende dall'ordine

```
……
inizializzazione
(condizioni iniziali)
 for (k=1; k <= n; k++) {
     calcolare il k-simo termine della
         successione
     aggiornare la "memoria"
 }
……
```

formule ricorrenti

$\Downarrow$

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, lineare a coefficienti non costanti

$$\boxed{y_k = y_{k-1} + k} \qquad \boxed{y_0 = 0}$$

genera la successione $\{y_k\}$ delle somme dei primi k numeri naturali $1,3,6,10,15,21,\ldots$

$$\boxed{y_k = \sum_{i=1}^{k} i = \frac{k(k+1)}{2}}$$

formule ricorrenti

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, lineare a coefficienti non costanti

```
s = s + k
```

$$y_k = y_{k-1} + k$$

$$y_0 = 0$$

$$y_k = 1 + 2 + 3 + 4 + \ldots + k = \sum_{i=1}^{k} i$$

$$y_k = (1 + 2 + 3 + 4 + \ldots (k-1)) + k = k + \underbrace{\boxed{\sum_{i=1}^{k-1} i}}_{y_{k-1}}$$

formule ricorrenti

$\Downarrow$

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, lineare a coefficienti non costanti

$$\boxed{y_k = y_{k-1} + a_k} \qquad \boxed{y_0 = 0}$$

genera la successione $\{y_k\}$ delle somme delle prime k componenti $a_1 + a_2 + a_3 + \ldots + a_k$ del vettore $\boldsymbol{a}$

formule ricorrenti

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, lineare a coefficienti non costanti

s = s + a(k)

$$y_k = y_{k-1} + a_k$$

$$y_0 = 0$$

$$y_k = \sum_{i=1}^{k} a_i$$

$$y_k = a_k + \boxed{\sum_{i=1}^{k-1} a_i}$$

$$y_{k-1}$$

formule ricorrenti

$\Downarrow$

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, lineare a coefficienti non costanti

$$y_k = k y_{k-1} \qquad y_0 = 1$$

genera la successione $\{y_k\}$ dei fattoriali dei primi k numeri naturali $1, 2, 6, 24, 120, 720, \ldots\ldots$

formule ricorrenti

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, lineare a coefficienti non costanti

```
f = f * k
```

$$y_k = k y_{k-1}$$

$$y_0 = 1$$

$$y_k = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots \cdot k$$

$$y_k = \underbrace{\boxed{\left(1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots \cdot (k-1)\right)}}_{y_{k-1}} \cdot k$$

formule ricorrenti

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, nonlineare a coefficienti non costanti

$$\boxed{y_k = \max\left(y_{k-1}, a_k\right)} \qquad \boxed{y_1 = a_1}$$

genera la successione $\{y_k\}$ dei massimi delle prime k componenti $a_1, a_2, a_3, \ldots, a_k$ del vettore $\boldsymbol{a}$

formule ricorrenti

consentono di descrivere **in modo formale** alcuni algoritmi basati sull'approccio incrementale

Esempio: la formula ricorrente del I ordine, nonlineare a coefficienti non costanti

```
if (a[k]>massimo)
   massimo = a[k];
```

$$y_k = \max\left(y_{k-1}, a_k\right)$$

$$y_1 = a_1$$

$$y_k = \max\left(a_1, a_2, a_3, \ldots, a_k\right)$$

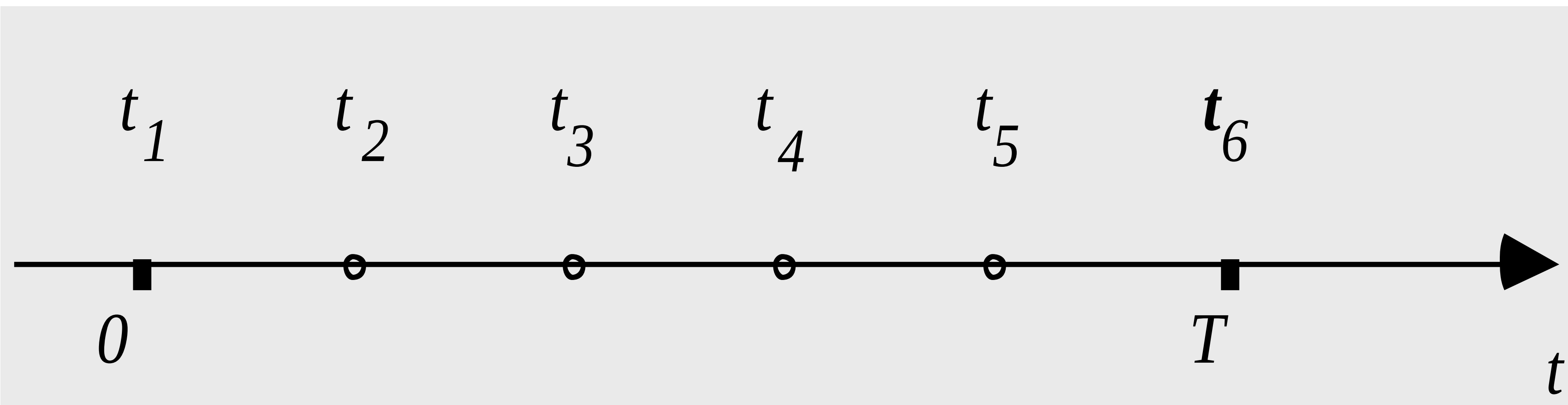
$$y_k = \max\left(\underbrace{\boxed{\max\left(a_1, a_2, a_3, \ldots, a_{k-1}\right)}}_{y_{k-1}}, a_k\right)$$

le **formule ricorrenti** (o **ricorrenze**)
sono anche dette
equazione alle differenze
oppure
sistema dinamico a tempo discreto

l'indice k può essere inteso come una **discretizzazione** del tempo

tempo discreto (intervallo di tempo discretizzato)

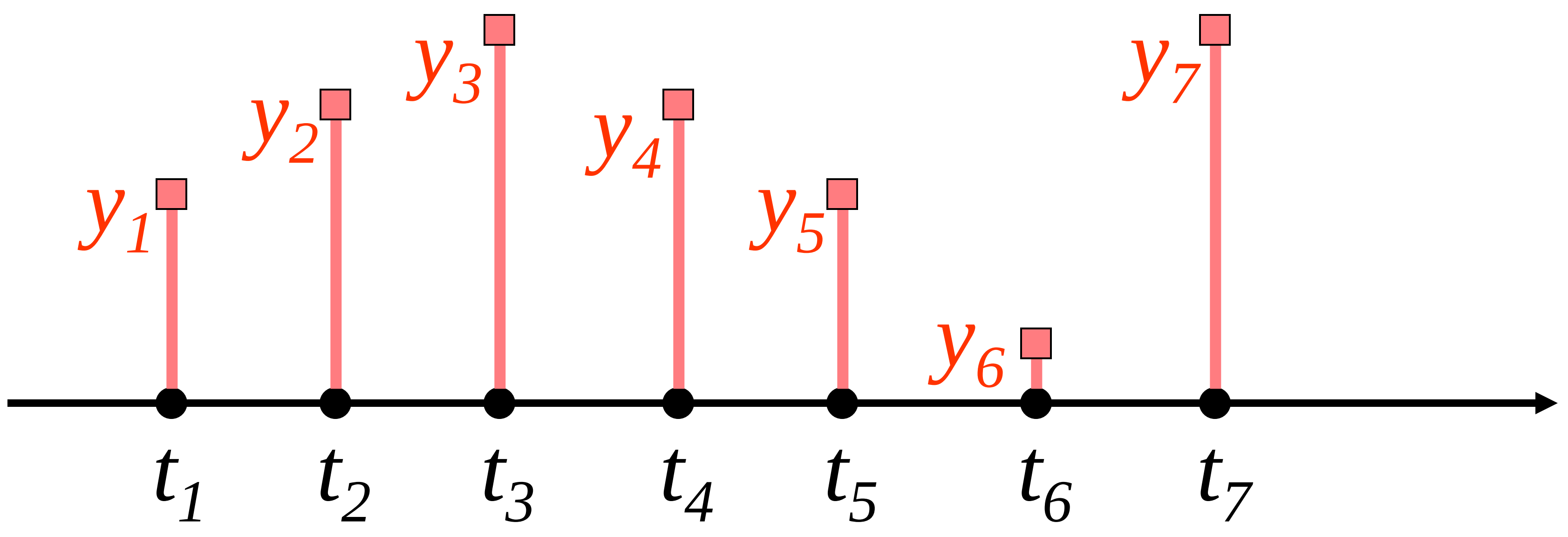
intervallo $[0,T]$ $\Rightarrow$ **griglia su** $[0,T]$



griglia su $[0,T]$: insieme di punti $\{t_1, t_2, t_3, t_4, t_5, t_6\}$

campionatura di una funzione su una **griglia**

y_k è il valore di una funzione $y = f\,(t)$
sul k-simo punto t_k della griglia



modello a tempo discreto di Malthus (dinamica di popolazioni)

idea:

una popolazione, in assenza di fattori che ne limitano la crescita, cresce in un dato periodo di tempo con una rapidità fissata e l'incremento è **proporzionale** al numero di individui

Thomas Robert Malthus (1766-1834) economista e demografo inglese

$$y_k = y_{k-1} + 0.5 y_{k-1}$$

la popolazione cresce del 50% in ogni unità di tempo se l'unità di tempo discreto (passo di griglia) è l'anno

la quantità 0.5 è il **tasso relativo di crescita** annuale della popolazione

y_k è la popolazione nell'anno k

y_0 è la popolazione iniziale

problema:
calcolare il saldo annuale di un conto bancario

- l'interesse percentuale annuale è 3%
- ogni anno viene depositata una quantità fissa b di denaro

la relazione che lega il saldo di un anno al saldo dell'anno precedente è la formula ricorrente

$$\boxed{y_k = (1 + p)y_{k-1} + b}$$

$$\boxed{y_k = (1 + 0.03)y_{k-1} + b}$$

Esempio: saldo dopo 5 anni, di un conto attivato con Euro10000, interesse del 3% e versamento annuale di Euro12000

```
main() {
float a,b,y_zero;
int n ;
 a = 1+0.03 ;
 b = 12E3 ;
 y_zero = 10E3 ;
 n = 5 ;
 printf (form_ric_1lc(n,a,b,y_zero));
 }
```

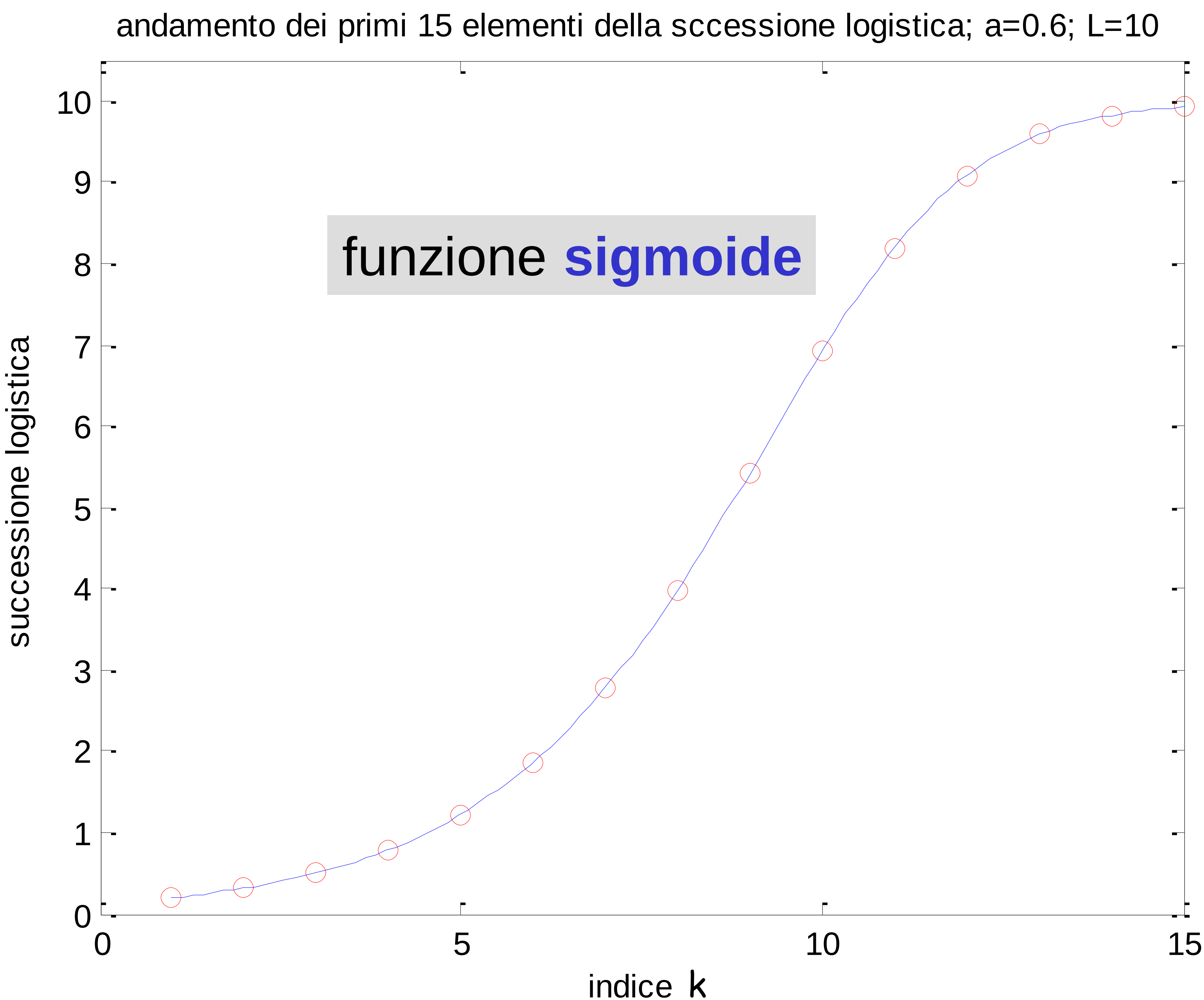
modello **logistico** (tempo discreto)

$$y_k = y_{k-1} + a y_{k-1}\left(1 - \frac{y_{k-1}}{L}\right)$$

- formula ricorrente **non lineare** del primo ordine a coefficienti costanti
- a = tasso relativo di crescita in assenza di fattori inibitori
- L = **popolazione limite** o **capacità portante**

$$y_k = y_{k-1} + a y_{k-1}\left(1 - \frac{y_{k-1}}{L}\right)$$

```
float formula_logistica(int n, float a, float L,
                                        float y_zero) {
float y;
int n, k;
 y = y_zero ;
 for (k=1; k<=n; k++) {
    y = y+a*y*(1.0-y/L) ;
  }
 return y ;
}
```


andamento dei primi 15 elementi della sccessione logistica; a=0.6; L=10
funzione sigmoide
successione logistica
indice k
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15

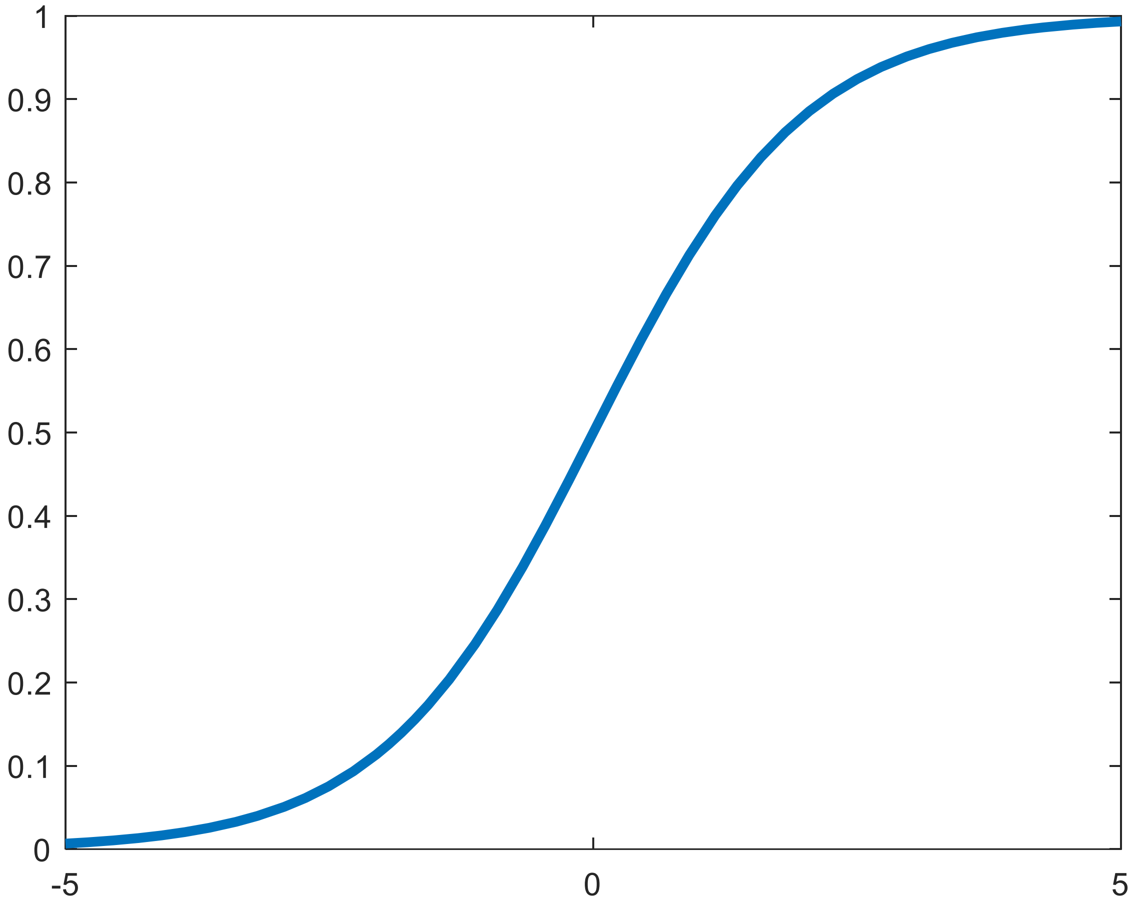


grafico della funzione **sigmoide** $y = \frac{1}{1+e^{-x}}$