

Esercizio 1 (3/7/2003)

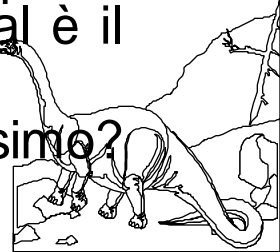
- Supponiamo che il sistema di paginazione utilizzato dal sistema operativo assegni un numero fissato di frame (blocchi di memoria) da 512B a ciascun processo e che l'algoritmo di sostituzione delle pagine sia LRU.

Prendiamo in considerazione il seguente programma:

```
...  
#define N 512  
int a[N];  
int i;  
  
...  
for (i=0; i < N/2; i++)  
    a[i] = a[2*i] + a[N-i-1];  
  
...
```

Si ponga attenzione solo ad `a` e si risponda ai seguenti quesiti:

- a) se la dimensione di un intero è 4B, qual è il numero minimo di frame-blocchi per i dati che minimizza il tasso di page fault?
- b) nel caso di minimo tasso di page fault, se il tempo medio di servizio di un page fault è di 25 millisecondi ($25 \cdot 10^{-3}$ sec) ed il tempo di accesso alla memoria di 100 microsecondi ($100 \cdot 10^{-6}$ sec), qual è il tempo di accesso effettivo della paginazione su richiesta?
- c) qual è la dimensione minima del TLB affinché l'hit ratio sia massimo?



Soluzione

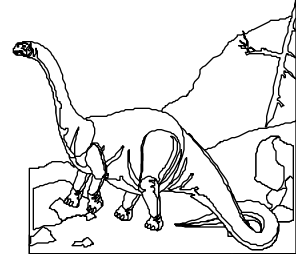
- Le pagine sono da 512B e ogni intero è 4B (128 elementi a pagina):

0	1	2	3
----------	----------	----------	----------

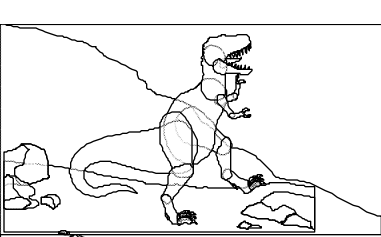
$a[0] \dots a[127]$ $a[128] \dots a[255]$ $a[256] \dots a[383]$ $a[384] \dots a[511]$

for (i=0; i < N/2; i++) a[i] = a[2*i] + a[N-i-1];

i = 0, ... 63	a[2*i] ->	0
	a[N-i-1] ->	3
	a[i] ->	0
i = 64, ... 127	a[2*i] ->	1
	a[N-i-1] ->	3
	a[i] ->	0
i = 128, ... 191	a[2*i] ->	2
	a[N-i-1] ->	2
	a[i] ->	1
i = 192, ... 255	a[2*i] ->	3
	a[N-i-1] ->	2
	a[i] ->	1



Soluzione



0	3	0	1	3	0	2	1	3	2	1
0	3	0	1	3	0	2	1	3	2	1
	0	3	0	1	3	0	2	1	3	2
			3	0	1	3	0	2	1	3
						1	3	0	0	0
inf	inf	2	inf	3	3	inf	4	4	3	3

$$C_1 = 0 \quad F_1 = 11$$

$$C_2 = 1 \quad F_2 = 10$$

$$C_3 = 4 \quad F_3 = 6$$

$$C_4 = 2 \quad F_4 = 4$$

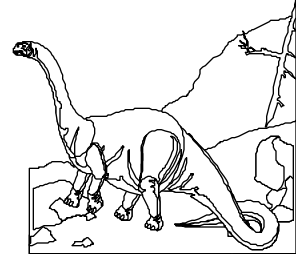
$$C_{inf} = 4$$

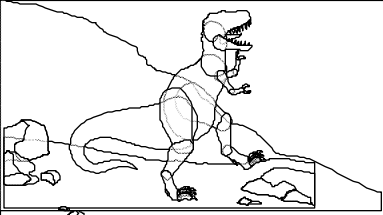
$$EAT = (1-p) m_a + p t_{pf}$$

$$EAT = 7/11 * 0,1ms + 4/11 * 25ms = 0.64 * 0,1ms + 0.36 * 25ms$$

$$= 9,064 \text{ ms}$$

**Hit ratio massimo con 4 blocchi.
Dimensione minima del TLB: 4.**





Esercizio 2 (21/7/2003)

- Supponiamo che il sistema di paginazione utilizzato dal sistema operativo assegni un numero fissato di frame da 512B a ciascun processo e che l'algoritmo di sostituzione delle pagine sia LRU. Prendiamo in considerazione il seguente programma:

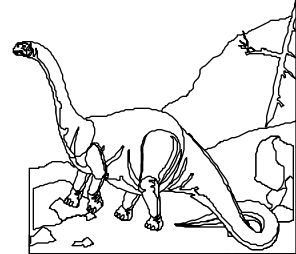
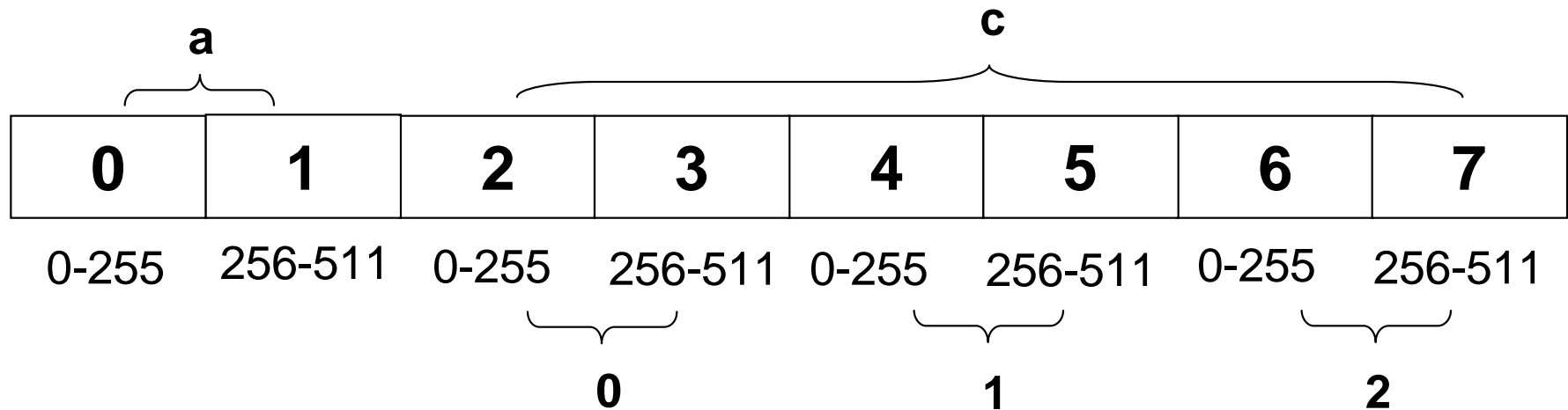
```
...
# define N 512
int a[N] ,c[ 3] [N] ;
int i,j ;
...
for ( i= 0; i < 3; i+ + )
for( j= 0; j< N/ 2; j+ + )
a[ 2* j ] = c[ i] [ j ] + c[ i] [ j+ N/ 2] ;
...
```

- Si ponga attenzione solo agli array a e c.
 - a) Se la dimensione di un intero è 2B, qual è il numero minimo di frame per i dati che minimizza il numero di page fault?
 - b) Nel caso di 3 blocchi, se il tempo medio di servizio di un page fault é di 40ms ($40 \cdot 10^{-3}$ sec) , il tempo di scrittura su disco delle pagine modificate 12ms ($12 \cdot 10^{-3}$ sec) ed il tempo di accesso alla memoria di 80microsecondi ($80 \cdot 10^{-6}$ sec), qual è il tempo di accesso effettivo della paginazione su richiesta?



Soluzione

- Le pagine sono da 512B e ogni intero è 2B (256 elementi/pagina)



Soluzione

2	3	0	2	3	1	4	5	0	4	5	1	6	7	0	6	7	1
2	3	0	2	3	1	4	5	0	4	5	1	6	7	0	6	7	1
	2	3	0	2	3	1	4	5	0	4	5	1	6	7	0	6	7
		2	3	0	2	3	1	4	5	0	4	5	1	6	7	0	7
Inf	Inf	Inf	3	3	inf	inf	inf	6	3	3	4	inf	inf	6	3	3	4

$$C_1 = 0 \quad F_1 = 18$$

$$C_2 = 0 \quad F_2 = 18$$

$$C_3 = 6 \quad F_3 = 12$$

$$C_4 = 2 \quad F_4 = 10$$

$$C_5 = 0 \quad F_5 = 10$$

$$C_6 = 2 \quad F_6 = 8$$

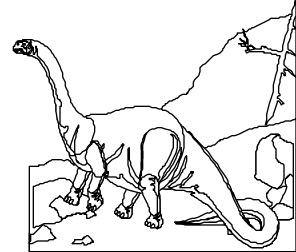
$$C_7 = 6$$

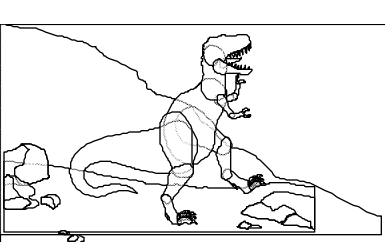
$$C_{inf} = 8$$

Hit ratio massimo con 6 blocchi.

$$EAT = 6/18 * 80 * 10^{-3} + 7/18 * (40 + 80 * 10^{-3}) + 5/18 * (40 + 80 * 10^{-3} + 12)$$

$$= 0,33 * 0,08 + 0,39 * 40,08 + 0,28 * 52,08 = 30,24ms$$





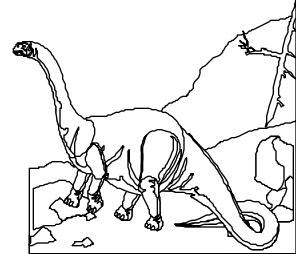
Esercizio 3 (10/9/2003)

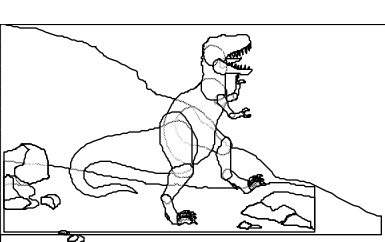
- Supponiamo che il sistema di paginazione utilizzato dal sistema operativo assegni un numero di 3 frame da 512B a ciascun processo.

Prendiamo in considerazione il seguente programma:

```
...
#define N 512
int A[N],C[N];
int i,j;
...
for (i=1; i <= 2; i++)
  for(j=0;j<N/2;j++)
    A[i*j] = A[2*i] + C[2*j];
...
```

- ☞ Si determini la stringa dei riferimenti alle pagine della memoria, considerando che un intero sia rappresentato con 2B.
- ☞ Si determini la tabella delle pagine, al termine dell'esecuzione della procedura, nel caso che l'algoritmo di sostituzione delle pagine sia (i) FCFS, (ii) LRU e (iii) CLOCK (all'inizio lancetta alla pagina 0).





Sostituzione della pagina

■ FCFS

0	2	0	3	0	2	0	3	1
0	2	2	3	3	3	3	3	1
	0	0	2	2	2	2	2	3
			0	0	0	0	0	2

4 page fault

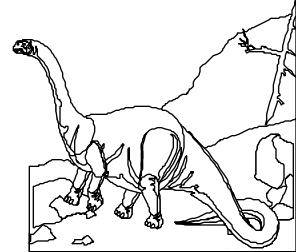
Pagina	Frame
0	-
1	0
2	2
3	1

■ LRU

0	2	0	3	0	2	0	3	1
0	2	0	3	0	2	0	3	1
	0	2	0	3	0	2	0	3
			2	2	3	3	2	0

4 page fault

Pagina	Frame
0	2
1	0
2	-
3	1



Sostituzione della pagina

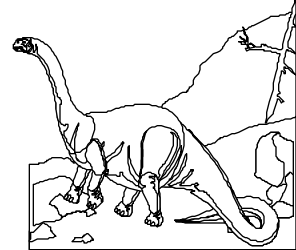
■ CLOCK

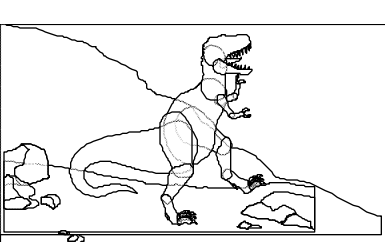
	0	2	0	3	0	2	0	3	1
→	0*	0*	0*	0*	0*	0*	0*	0*	1*
		2*	2*	2*	2*	2*	2*	2*	2
			3*	3*	3*	3*	3*	3*	3

← 4 page fault

☞ La tabella delle pagine al termine dell'esecuzione è:

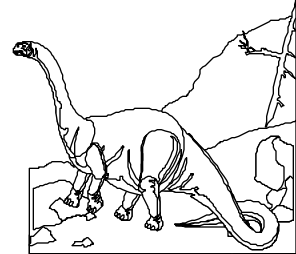
Pagina	Frame	R
0	-	0
1	0	1
2	1	0
3	2	0





Esercizio 4 (30/9/2003)

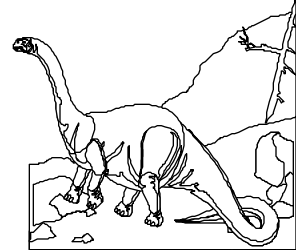
- Considerate la seguente successione di riferimenti di pagine:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 3, 2, 7, 6, 3, 2, 1, 2, 3, 6.
- a. Dite quante assenze di pagina (page fault) avrebbero luogo usando 5, 6 o 7 blocchi (frame) di memoria, con i seguenti algoritmi di sostituzione:
 - i. LRU,
 - ii. FIFO,
 - iii. Orologio (CLOCK) con lancetta inizialmente sul 1° elemento della tabella delle pagine.
- b. Determinare il tempo di accesso effettivo della paginazione su richiesta per LRU con 5 blocchi, se:
 - i. il tempo medio di servizio di un page fault in lettura è di 80 millisecondi ($80 \cdot 10^{-3}$ sec),
 - ii. il tempo medio di servizio di un page fault in scrittura è di 140 millisecondi ($140 \cdot 10^{-3}$ sec) ed
 - iii. il tempo di accesso alla memoria è di 80 microsecondi ($80 \cdot 10^{-6}$ sec), nell'ipotesi che l'accesso alle pagine 1, 2, 3 sia sempre in scrittura.



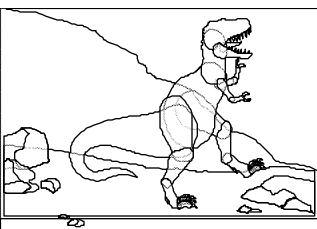
Soluzione

$$\text{LRU}(5) = 8 \quad \text{LRU}(6) = 7 \quad \text{LRU}(7) = 7$$

1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
	1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3
		1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	3	1	2
			1	1	3	4	2	1	5	6	6	1	3	2	7	6	6	6	1
						3	4	4	4	5	5	6	1	1	1	7	7	7	7
							3	3	3	4	4	5	5	5	5	5	5	5	5
												4	4	4	4	4	4	4	4
∞	∞	∞	∞	3	4	∞	∞	4	4	6	3	∞	5	4	4	5	2	3	4



Soluzione



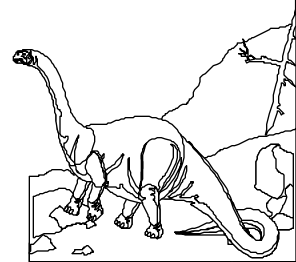
FIFO(5) = 10

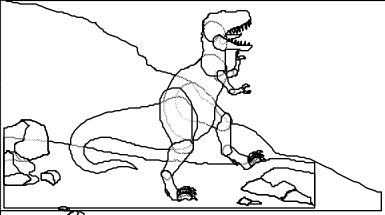
1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
1	2	3	4	4	4	5	6	6	1	1	2	7	7	3	3	3	3	3	3
	1	2	3	3	3	4	5	5	6	6	1	2	2	7	7	7	7	7	7
		1	2	2	2	3	4	4	5	5	6	1	1	2	2	2	2	2	2
			1	1	1	2	3	3	4	4	5	6	6	1	1	1	1	1	1
						1	2	2	3	3	4	5	5	6	6	6	6	6	6
P	P	P	P			P	P		P		P	P		P					

FIFO(6) = 10

1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
1	2	3	4	4	4	5	6	6	6	6	6	7	7	7	7	1	2	3	3
	1	2	3	3	3	4	5	5	5	5	5	6	6	6	6	7	1	2	2
		1	2	2	2	3	4	4	4	4	4	5	5	5	5	6	7	1	1
			1	1	1	2	3	3	3	3	3	4	4	4	4	5	6	7	7
							1	1	1	1	1	2	2	2	2	3	4	5	5
P	P	P	P			P	P					P				P	P	P	

FIFO(7) = 7

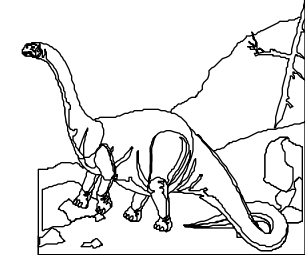


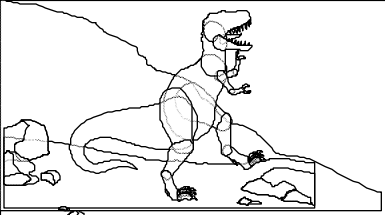


Soluzione

CLOCK(5)= 8

1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
1*	1*	1*	1*	1*	1*	1*	6*	6*	6*	6*	6*	6*	6*	6*	6*	6*	6*	6*	6*
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	2*	2*	2*	2*	2*	2*	2	2*	2	2	2*	2*	2*	2*	2*	2*	2*	2*	2*
							-	-											
		3*	3*	3*	3*	3*	3	3	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*
			4*	4*	4*	4*	4	4	4	3*	3*	3*	3*	3*	3*	3*	3*	3*	3*
							5*	5	5	5	5	5	7*	7*	7*	7*	7*	7*	7*
P	P	P	P				P		P	P		P							

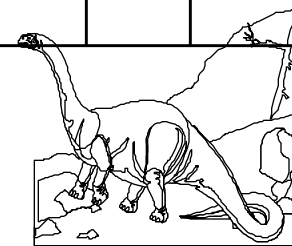


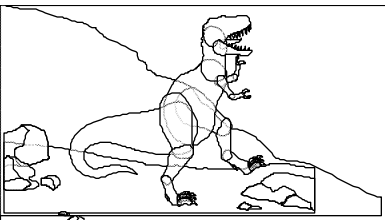


Soluzione

CLOCK(6)= 6

1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	7*	7*	7*	7*	7*	7*	7*	7*
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*	-	-	-	-
	2*	2*	2*	2*	2*	2*	2*	2*	2*	2*	2*	2	2	2	2	2	2*	2*	2*
												-	-	-	*	-	-	-	-
		3*	3*	3*	3*	3*	3*	3*	3*	3*	3*	3	3	3*	3	3	3	3*	3*
															*				
			4*	4*	4*	4*	4*	4*	4*	4*	4*	4	4	4	4	1*	1*	1*	1*
						5*	5*	5*	5*	5*	5*	5	5	5	5	5	5	5	5
															-	-	-	-	-
							6*	6*	6*	6*	6*	6	6*	6*	6	6*	6*	6*	6*
															*				
P	P	P	P									P	P						

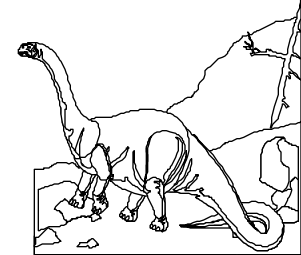




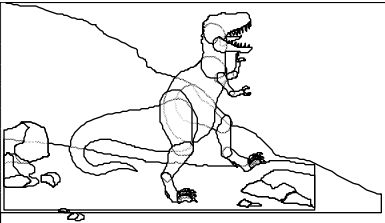
Soluzione

LRU(5) = 8

1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3	6
	1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	2	1	2	3
		1	2	3	4	2	1	5	6	2	1	3	2	7	6	3	3	4	2
			1	1	3	4	2	1	5	6	6	1	3	2	7	6	6	6	4
						3*	4	4	4	5	5	6	1	1	1	7	7	7	7
P	P	P	P			P	P			P		P							

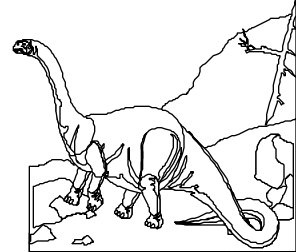


Soluzione

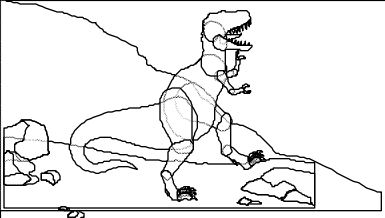


- $t_{pf} = 80\text{ms}$,
- $t_{so} = 140\text{ms}$,
- $\gamma = 0.08\text{ms}$,
- $p_1 = 12/20$ (probabilità che non si verifichi un page fault),
- $p_2 = 7/20$ (probabilità che si verifichi un page fault senza scrittura della pagina sostituita),
- $p_3 = 1/20$ (probabilità che si verifichi un page fault con scrittura della pagina sostituita)

$$\begin{aligned} EAT &= 12/20 * 80 + 7/20 [80 + 0.08] + 1/20 [80 + 140 + 0.08] = \\ &= 0,6 * 80 + 0,35 * 80,08 + 0,05 * 220,08 = 87,032 \end{aligned}$$



Esercizio 5

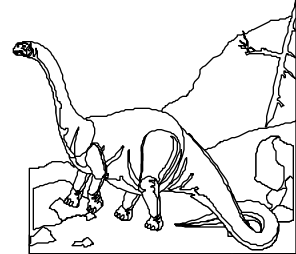


- Da un'analisi delle prestazioni della memoria virtuale, con paginazione su richiesta che utilizzi un numero fissato di blocchi e si avvalga di un TLB, si individuino i seguenti parametri caratteristici:
 - ☞ γ tempo di accesso alla memoria fisica
 - ☞ ε tempo di accesso al TLB
 - ☞ t_{pf} tempo di servizio del page fault
 - ☞ t_{so} tempo di swap-out (scrittura della pagina a seguito di un'eccezione)
 - ☞ p_1 probabilità che non si verifichi un page fault
 - ☞ p_2 probabilità che si verifichi un page fault senza scrittura della pagina sostituita
 - ☞ p_3 probabilità che si verifichi un page fault con scrittura della pagina sostituita
 - ☞ α probabilità di successo di un accesso al TLB

Qual è il tempo effettivo di accesso alla memoria?

$$\begin{aligned} EAT = & p_1 [(1-\alpha) (\varepsilon + \gamma) + \alpha (\varepsilon + \gamma) + \gamma] + \\ & p_2 [(1-\alpha) (\varepsilon + \gamma) + t_{pf} + \gamma] + \\ & p_3 [(1-\alpha) (\varepsilon + \gamma) + t_{pf} + t_{so} + \gamma]. \end{aligned}$$

NB: $p_1 + p_2 + p_3 = 1$



Esami

■ Date:

☞ 23 giugno 2004

☞ 6 luglio 2004

