

# Informatica 3

Prof. Giovanni Agosta  
Prof. Alessandro Campi  
Prof. Maristella Matera

## Prova scritta – Appello del 3/07/2009<sup>1</sup>

COGNOME e NOME:..... Matricola:.....  
SEZIONE: Agosta  Campi  Matera

---

<sup>1</sup>Tempo: 1,5 ore. Possono essere consultati libri di testo e slide del corso. È consentito scrivere a matita. Scrivere il proprio nome sugli eventuali fogli aggiuntivi. *Rispondere punto per punto alle domande!*

## **Esercizio 1 (7 punti)**

Progettare (in Python o in Java) una funzione che implementi l'operazione di prodotto tra matrici parallelizzando il calcolo su più thread. Discutere le ragioni delle scelte fatte nella parallelizzazione.

## Esercizio 2 (7 punti)

Sia dato il seguente programma Python:

```
#!/usr/bin/python
from sys import argv
x= [ int(i) for i in argv[1] ]
print len(argv), x[-1]

def f(a) :
    return len(a), reduce(lambda x, y : x+y, a)

print f(x)
print max([ n**2 for n in x if n>0 and n<7 ])

def h(a):
    if not len(a) : return a
    return [ i for i in a[1:] if i< a[0] ] + \
           [a[0]] + [ i for i in a[1:] if i>= a[0] ]

print h(x)
```

Notate che il carattere `\` indica che il programma prosegue alla linea successiva, ignorando il carattere “a capo”.

Rispondere alle seguenti domande, giustificando la risposta:

1. Cosa stampa a video il programma, se viene mandato in esecuzione passando come parametro dalla linea di comando il vostro numero di matricola?

2. Cosa stampa a video il programma, se non viene passato alcun parametro dalla linea di comando?

### Esercizio 3: Hashing (7 punti)

**Quesito 1 (3 Punti)** Data una tabella hash di 13 elementi, inserire le chiavi sotto riportate usando la funzione di hash  $h = key \bmod 13$  e il *double hashing* per la gestione delle collisioni ( $h2 = 1 + (key \bmod 5)$ ).

1, 24, 8, 10, 12, 21, 47, 34

**Quesito 2 (4 Punti)** Data una tabella hash di 17 elementi, inserire le chiavi sotto riportate usando la funzione di hash  $h = key \bmod 17$  e il *linear probing* per la gestione delle collisioni.

2, 5, 14, 31, 15, 32, 48, 49.

Dopo l'inserimento dell'ultimo elemento si indichi, per ognuno degli slot liberi, la probabilità che l'inserimento successivo venga effettuato in quello slot.

## Esercizio 4: Algoritmi (7 punti)

**Quesito 1 (2 Punti)** Data una sequenza di  $n+1$  numeri interi  $a_1, a_2, \dots, a_n$ , si vuole calcolare la somma  $b = a_1 + a_2 + \dots + a_n$ . Definire (in Python, Java o C) un algoritmo del tipo *divide et impera* in grado di risolvere questo problema e discuterne la complessità.

**Quesito 2 (5 Punti)** Sia  $M$  una matrice di  $n \times n$  interi e  $k$  un valore intero da ricercare nella matrice. Gli elementi di tutte le righe sono ordinati in modo crescente. Gli elementi di ogni colonna sono ordinati in modo crescente. Progettare (in Python, Java o C) un algoritmo per determinare se  $k$  è contenuto nella matrice e discuterne la complessità.

### Esercizio 5: Complessità (5 punti)

Determinare i bound asintotici di complessità più stretti per le seguenti ricorrenze:

1.  $T(n) = 3T(n/4) + n^5$
2.  $T(n+2) = 2T(n+1) + 3T(n)$ , con  $T(0) = 0$  e  $T(1) = 1$
3.  $T(n) = 3T(n/3) + 2n/(\log_3 n)$

Per l'ultimo caso, si ricorda che  $H_m = \sum_{i=1}^m 1/i$  è limitata da  $\ln(m+1) \leq H_m \leq (\ln m) + 1$ .