

**Ingegneria degli Algoritmi (A.A. 2010-2011)**

Corsi di Laurea in Ingegneria Informatica e Automatica, Ingegneria dei Sistemi Informatici, e Laurea Magistrale in Ingegneria Informatica

*Sapienza Università di Roma*

**Secondo appello (14/7/2011) – Durata 2 ore – 5/6 cfu**

<p>Cognome: _____</p> <p>Nome: _____</p> <p>Matricola: _____</p>	<p>Autorizzo la pubblicazione del voto di questo esame sul sito web <a href="http://www.dis.uniroma1.it/~demetres/didattica/ae">http://www.dis.uniroma1.it/~demetres/didattica/ae</a>, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede,</p> <p>_____</p>
--	---

Rispondere alle seguenti domande, **motivando le risposte** (risposte non motivate saranno considerate nulle).

---

**Domanda 1 [8 punti]**

Si illustrino le seguenti tecniche di ottimizzazione dei programmi, discutendone funzionamento e obiettivi, e fornendo degli esempi di applicazione:

1. Strength reduction
2. Common subexpression elimination
3. Constant propagation
4. Loop unrolling

---

**Domanda 2 [8 punti]**

Si consideri un progetto C formato dai seguenti file posizionati nelle due directory `test/include` e `test/src`:

1. `test/include/uno.h`
2. `test/src/uno.c` che include `uno.h`
3. `test/include/due.h` che include `uno.h`
4. `test/src/due.c` che include `due.h`
5. `test/src/main.c` che include `uno.h` e `due.h` e fornisce la funzione `main`

Scrivere un file `test/Makefile` per compilare il progetto mediante il comando `make`, in modo che il file eseguibile prodotto venga creato nella directory `test/bin`, già esistente. Scrivere il `makefile` in modo che una qualsiasi modifica a uno dei cinque file che costituiscono il progetto implichi il minimo numero di operazioni di ricompilazione. Usare opportune definizioni di macro per rendere più

compatto il codice. Assumere di usare il compilatore `gcc` su una macchina Linux/Unix.

---

**Domanda 3 [8 punti]**

Si illustri un algoritmo efficiente per moltiplicare due matrici quadrate memorizzate in memoria esterna e si analizzi il numero di I/O richiesti dall'algoritmo usando la notazione asintotica.

---

**Domanda 4 [8 punti]**

Si progetti una struttura dati con le seguenti operazioni:

1. `insert(x)`: aggiunge un elemento `x` alla struttura;
2. `query(i)`: restituisce l' $i$ -esimo elemento inserito, con  $1 \leq i \leq n$ , dove  $n$  è il numero corrente di elementi della struttura.

Si consideri come esempio la seguente sequenza mista di operazioni:

- `insert(17)`: aggiunge 17 alla struttura
- `insert(31)`: aggiunge 31 alla struttura
- `query(2)`: restituisce 31
- `insert(62)`: aggiunge 62 alla struttura
- `query(1)`: restituisce 17
- `query(3)`: restituisce 62

Tutte le operazioni devono avere costo ammortizzato costante su qualsiasi sequenza mista di operazioni `insert` e `query`. Inoltre, lo spazio utilizzato deve essere  $O(n)$ , dove  $n$  è il numero di elementi nella struttura. Analizzare il costo delle operazioni mediante il metodo dei crediti.