

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 (23/9/2011) – Durata 2 ore – 5/6 cfu

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

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

Domanda 1 [8 punti]

Si consideri il seguente programma C:

```
#include <stdio.h>

#define max(a, b) ((a)>(b) ? (a):(b))

int main() {
    int x = 10, y = 20;
    int z = max(x, y++);
    printf("x=%d y=%d z=%d\n", x, y, z);
    return 0;
}
```

Dire cosa stampa il programma e perché.

Domanda 2 [8 punti]

Si risponda alle seguenti due domande sulla misurazione delle prestazioni di un programma:

1. Si discuta la differenza fra i tempi *real*, *user* e *system*. Si descriva inoltre la tecnica dell'*interval counting*. [4 punti]
2. Si enunci la legge di Amdahl, illustrandone l'applicazione con un esempio a scelta. [4 punti]

Domanda 3 [8 punti]

Si illustri un algoritmo efficiente per ordinare una sequenza di elementi memorizzati 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 variante della struttura dati *pila*, realizzata mediante una *lista collegata semplice*, che fornisca le seguenti operazioni:

1. `push(x)`: aggiunge un elemento x in cima alla pila;
2. `multi-pop(k)`: elimina i k elementi più in cima alla pila, assumendo che k sia minore o uguale al numero di elementi nella pila.
3. `top()`: restituisce l'elemento corrente in cima alla pila, assumendo che la pila non sia vuota.

Si consideri come esempio la seguente sequenza mista di operazioni:

- `push(17)`: aggiunge 17 in cima alla pila
- `push(31)`: aggiunge 31 in cima alla pila
- `push(12)`: aggiunge 12 in cima alla pila
- `push(23)`: aggiunge 23 in cima alla pila
- `multi-pop(3)`: elimina i tre elementi 23, 12 e 31 dalla cima della pila
- `top()`: restituisce 17

Tutte le operazioni devono avere costo ammortizzato $O(1)$ su qualsiasi sequenza valida e mista di operazioni `push`, `multi-pop` e `top`. Inoltre, lo spazio utilizzato deve essere $O(n)$, dove n è il numero di elementi nella pila. Analizzare il costo delle operazioni mediante il metodo dei crediti.