

INFORMATICA GENERALE

Primo Esonero

docenti: TIZIANA CALAMONERI, IVANO SALVO
Sapienza Università di Roma

29 aprile 2014

Esercizio 1 (10 punti) Si risolva la seguente equazione di ricorrenza:

$$T(n) = 4T(n/2) + \Theta(\log n)$$

$$T(1) = \Theta(1)$$

utilizzando:

- (4 punti) il metodo iterativo;
- (3 punti) il metodo dell'albero;
- (3 punti) il metodo principale.

In tutti i casi è necessario dettagliare il procedimento usato e giustificare le proprie risposte.

Esercizio 2 (10 punti) Si supponga di avere a disposizione un esecutore la cui unica abilità aritmetica sia la seguente funzione:

```
int scomponi(int n, int *f1, int *f2)
```

che ha il seguente comportamento: sotto la preconditione che il parametro n sia un numero intero positivo restituisce come risultato 0 se n contiene un numero primo, e 1 se n contiene un numero composto. In tal caso, carica nei parametri $f1$ ed $f2$ due fattori (maggiori strettamente di 1) il cui prodotto è il valore passato nel parametro n .

Ad esempio, siano x ed y due variabili intere. La chiamata `scomponi(17, &x, &y)` restituisce 0. Viceversa la chiamata `scomponi(24, &x, &y)` restituisce 1 e carica x ed y con due fattori di 24, per esempio 4 e 6, oppure 8 e 3, oppure 12 e 2 (ma non possiamo fare ipotesi su quali essi siano).

1. Scrivere una funzione *ricorsiva* `maxPrimoRec(int m)` che usa `scomponi` per calcolare il massimo fattore primo di un numero positivo `m`;
2. ★scrivere una funzione *iterativa* `maxPrimoIt(int m)` che usa `scomponi` per calcolare il massimo fattore primo di un numero positivo `m` [SUGGERIMENTO: aiutarsi con un array ausiliario dove memorizzare i fattori via via calcolati].

Ovviamente, potete usare gli operatori relazionali `<`, `<=` e `==` tra interi. In entrambi i casi, cercate di argomentare (anche informalmente) sulla terminazione e sulla correttezza delle funzioni scritte.

Esercizio 3 (10 punti) Si consideri un vettore non ordinato $A[1..n]$ contenente n numeri interi distinti e sia dato un intero k , $1 \leq k \leq n$.

Si progetti un algoritmo (il più efficiente possibile) che, presi in input A e k , stampi i k valori più piccoli presenti in A . Di tale algoritmo:

- a. (**3 punti**) si dia la spiegazione a parole;
- b. (**5 punti**) si scriva lo pseudocodice;
- c. (**2 punti**) si calcoli il costo computazionale.

Se ritenuto necessario, è possibile fare uso anche di funzioni ausiliarie viste a lezione senza bisogno di scriverne lo pseudocodice, purché se ne specifichi chiaramente l'input e l'output.