

INFORMATICA GENERALE

Compito Scritto

docenti: TIZIANA CALAMONERI, IVANO SALVO
Sapienza Università di Roma

10 Settembre 2012

Esercizio 1 (10punti) Si consideri il seguente frammento di codice:

```
funzione(n)
if n == 0 || n == 1
    return 0
return funzione(n/4)+fun(n)
```

1. **(2 punti)** Si scriva l'equazione di ricorrenza che descrive la complessità della funzione sopra definita nell'ipotesi che la funzione `fun` abbia tempo di esecuzione nel caso peggiore $\Theta(n)$;
2. **(4punti)** Si risolva l'equazione di ricorrenza trovata con due metodi diversi, giustificandone i passaggi;
3. **(4 punti)** Come cambia l'equazione di ricorrenza, e la relativa soluzione, nel caso in cui il tempo di esecuzione della funzione `fun` sia $O(n)$? E come nel caso in cui sia $\Omega(n)$? È possibile che, nel primo caso ($O(n)$), la complessità di `funzione` sia $\Theta(\log n)$? se sì, quale complessità dovrebbe avere la funzione `fun`?

Esercizio 2 (10 punti) Sia P una lista non vuota che contiene ordinatamente i primi k numeri primi ($k > 0$). Scrivere il codice di una funzione `C` di prototipo `int ennesimoPrimo(lista P, int n)`; che restituisce l'ennesimo numero primo. Se $n \leq k$, si limita a ritornare il contenuto dell' n -esimo nodo di P . Altrimenti, oltre a ritornare l' n -esimo numero primo, modifica P aggiungendovi in coda altri $n - k$ numeri primi.

Esercizio 3 (10 punti) Un albero si dice *albero binario di ricerca opposto* se, per ogni nodo v , tutti i nodi nel sottoalbero sinistro di v hanno chiave \geq della chiave di v e tutti i nodi nel sottoalbero destro di v hanno chiave $<$ della chiave di v .

Progettare un algoritmo che, dato un albero binario di ricerca T_1 , lo trasformi in un albero binario di ricerca opposto T_2 . Dell'algoritmo proposto:

1. **(4 punti)** si dia la descrizione a parole;
2. **(3 punti)** si scriva lo pseudocodice;
3. **(3 punti)** si valuti la complessità.