

Secondo Esonero e Compito Scritto – 20 Giugno 2011

Si fa presente che chi volesse fare il secondo esonero deve risolvere solo la seconda parte nel tempo massimo di 1 ora e mezza; chi volesse fare il compito per intero ha un tempo massimo di 2 ore e mezza.

Nel momento in cui venga consegnato il compito intero, l'eventuale voto del primo esonero verrà automaticamente cancellato.

PRIMA PARTE

ESERCIZIO 1. (7 punti)

Si consideri la seguente funzione C:

```
int funzione (tree_pointer primo, tree_pointer secondo)
{
    if (!primo && !secondo)
        return 1
    else return((primo && secondo) &&
                (primo->dati==secondo->dati) &&
                funzione(primo->fsin, secondo->fsin) &&
                funzione(primo->fdes,secondo->fdes) )
}
```

dove tree_pointer è un puntatore ad una struttura composta di una porzione dati e due puntatori dello stesso tipo fsin ed fdes ed, all'inizio, la funzione viene richiamata in modo tale che i due puntatori primo e secondo puntino alle radici di due alberi binari, entrambi di n nodi.

- (2 punti)** Spiegare cosa restituisce la funzione
- (2 punti)** Scrivere l'equazione di ricorrenza che definisce la complessità computazionale della funzione
- (3 punti)** Risolvere l'equazione trovata nel caso in cui:
 - i due alberi siano uguali e completi
 - i due alberi siano uguali e rappresentanti due cammini

ESERCIZIO 2. (8 punti)

Dato un vettore di n numeri reali non nulli, progettare un algoritmo efficiente che posizioni tutti gli elementi negativi prima di tutti gli elementi positivi.

Dell'algoritmo progettato si dia:

- (3 punti)** la descrizione a parole
- (3 punti)** lo pseudocodice
- (2 punti)** la complessità computazionale.

SECONDA PARTE

ESERCIZIO 3. (8 punti)

Sia dato un albero binario di ricerca T . Sia v un nodo interno di T , e sia T_v il sottoalbero con radice v . Si progetti un algoritmo efficiente che, ricevuto in input un nodo $v \in T$ e una chiave k , restituisca il numero di nodi in T_v con chiave $\leq k$;

Dell'algoritmo progettato si dia:

- (3 punti)** la descrizione a parole
- (3 punti)** lo pseudocodice
- (2 punti)** la complessità computazionale.

ESERCIZIO 4. (7 punti)

Sia dato un grafo G con n nodi ed m archi composto di k componenti connesse.

- (2 punti)** Quanti alberi genera una visita in profondità che visiti tutti i nodi? e quanti una visita in ampiezza?
- (2 punti)** Quanti archi saranno classificati come "arco dell'albero" da una visita in profondità? quanti come "arco di riporto"? e quanti come "arco di attraversamento"?
- (3 punti)** La relazione "i nodi v e w sono nella stessa componente connessa" è una relazione di equivalenza?

Per ciascun quesito motivare la risposta.