

**PROVA D'ESAME DI INFORMATICA II**  
**21 LUGLIO 2008**

CORSO DI LAUREA IN MATEMATICA  
UNIVERSITÀ DEGLI STUDI DI UDINE

**Esercizio 1 - Punti 3.** Valutare il comportamento asintotico della seguente funzione ricorsiva

$$T(n) = \begin{cases} 3T(n/2) + \sqrt{n} & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

**Esercizio 2 - Punti 5.** Scrivere con criterio di costo logaritmico l'equazione ricorsiva di complessità in termini di tempo della seguente funzione

```
int Wot(int a){
    int c, x, i;
    if(a==0){
        return 3;
    }
    else{
        c=Wot(a-1);
        x=1;
        for(i=0;i<c;i++){
            x=x*3;
        }
        return x;
    }
}
```

Quale è l'output della funzione `Wot` su un generico input intero positivo?

**Esercizio 3 - Punti 5.** Dato un vettore  $A$  di interi di lunghezza  $n$ , diciamo che  $A$  contiene l'intervallo  $[a, b]$  se tutti i numeri interi compresi tra  $a$  e  $b$  occorrono in  $A$  ( $a$  e  $b$  sono interi). Si vuole determinare l'intervallo di cardinalità massima contenuto nel vettore  $A$ . Scrivere il codice di una funzione  $C$  per risolvere tale problema. Calcolarne la complessità con criterio di costo uniforme.

**Esercizio 4 - Punti 5.** Un albero capovolto è un albero in cui ogni nodo ha una chiave intera ed un puntatore al proprio genitore. In questo modo ogni nodo può avere un numero arbitrario di figli.

Si scriva in  $C$  la struttura dati albero capovolto.

Dato un nodo di un albero capovolto (un puntatore al nodo) si vuole determinare la radice dell'albero. Si scriva in  $C$  una funzione che risolva tale problema. Se ne calcoli la complessità con criterio di costo uniforme.

Si modifichi la funzione appena descritta per fare in modo che al termine della sua esecuzione il nodo di partenza e tutti i suoi antenati (suo padre, il padre di

suo padre, ...) siano diventati figli della radice. Se possibile, risolvere il problema “passando” da ogni nodo al più una volta.

**Esercizio 5 - Punti 6.** Realizzare la struttura dati grafo con liste di adiacenza. Sia  $G = (N, E)$  un grafo non orientato.  $G$  si dice bipartito se esistono due insiemi  $A, B \subseteq N$  tali che  $A \cup B = N$ ,  $A \cap B = \emptyset$  e per ogni  $\{u, v\} \in E$  vale che  $u \in A$  e  $v \in B$  oppure  $v \in A$  e  $u \in B$ .

Scrivere una funzione  $C$  che permette di stabilire se un grafo è bipartito. Calcolarne la complessità con criterio di costo uniforme.

**Esercizio 6 - Punti 6.** Si consideri una nozione restrittiva di macchina di Turing ottenuta impedendo la sovrascrittura del simbolo ‘blank’  $\sqcup$  (vale a dire, senza transizioni del tipo  $(s, \sqcup, s', a, d)$  con  $a \neq \sqcup$ ).

Dire se tale nozione di macchina è un modello di calcolo universale, vale a dire, equivalente a quello delle macchine di Turing standard o delle macchine URM. In caso di risposta affermativa, descrivere a parole un modo per simulare ogni possibile macchina di Turing con una macchina che non sovrascrive mai il simbolo  $\sqcup$ . In caso di risposta negativa, fornire un esempio di macchina di Turing che non pu essere simulata nel modello ristretto di calcolo considerato.