

**Algoritmi e Strutture Dati – 03/02/14****Esercizio 1 – Punti  $\geq 8$  (Parte A)**

Trovare un limite superiore per la seguente equazione di ricorrenza utilizzando il metodo per tentativi (altrimenti detto per sostituzione).

$$T(n) = \begin{cases} \left( \sum_{i=1}^{\lfloor \log n \rfloor} T(n/2^i) \right) + 1 & n > 1 \\ 1 & n = 1 \end{cases}$$

**Esercizio 2 – Punti  $\geq 7$  (Parte A)**

Si consideri un albero binario  $T$  su cui viene eseguito il seguente meccanismo di comunicazione. Il nodo radice  $r$  di  $T$  deve consegnare un messaggio  $m$  a tutti i nodi di  $T$ . La comunicazione avviene a turni: in ogni turno, un nodo può inoltrare  $m$  al massimo ad un altro nodo scelto fra i propri figli, e solo se ha ricevuto  $m$  in un turno precedente. In un dato turno, più nodi possono spedire contemporaneamente. La comunicazione termina quando tutti i nodi hanno ricevuto il messaggio.

Scrivere un algoritmo che calcola il numero minimo di turni necessari per consegnare il messaggio a tutti i nodi.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

**Esercizio 3 – Punti  $\geq 6$  (Parte A)**

Scrivere un algoritmo che, dato un grafo non orientato  $G = (V, E)$  connesso, restituisce un nodo tale che, se rimosso, lascia il grafo connesso.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.

**Esercizio 4 – Punti  $\geq 12$  (Parte B)**

Si consideri una scacchiera  $S$  di dimensione 4 righe (numerata  $0 \dots 3$ ) per  $n$  colonne. In ogni cella della scacchiera è iscritto un numero intero positivo, che rappresenta un guadagno. Si consideri il problema di piazzare 4 torri su tale scacchiera, in modo tale che

- Le torri non si minaccino l'un l'altra, ovvero che non siano sulla stessa riga o sulla stessa colonna.
- La somma dei guadagni contenuti nelle celle occupate dalle torri è massima

Scrivere un algoritmo che risolva tale problema.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale.