

Algoritmi e Principi dell'Informatica

Appello del 29 settembre 2017

Chi deve sostenere l'esame integrato (API) deve svolgere tutti gli esercizi in 2 ore.

Chi deve sostenere solo il modulo di *Informatica teorica* deve svolgere gli Esercizi 1 e 2 in 1 ora.

Chi deve sostenere solo il modulo di *Informatica 3* deve svolgere gli Esercizi 3 e 4 in 1 ora.

NB: i punti attribuiti ai singoli esercizi hanno senso solo con riferimento all'esame integrato e hanno valore puramente indicativo.

Esercizio 1 (8 punti)

Si specifichi in logica del prim'ordine la funzione unaria f , definita per tutti gli interi non negativi: detto n l'argomento della funzione f , essa restituisce la somma degli interi da 1 a n . Ad esempio, $f(5)$ vale 15 (cioè $1+2+3+4+5$).

Oltre ai connettivi logici, quantificatori e punteggiatura consentiti nella logica del prim'ordine, nella specifica si può fare uso *esclusivamente* dei seguenti simboli:

- La funzione binaria di somma (da indicarsi con il consueto simbolo $+$ e notazione infissa);
- La funzione binaria di sottrazione (come sopra, con il simbolo $-$);
- I predicati binari '=' , di uguaglianza e '>' , di confronto tra interi;
- Le costanti 0 e 1.

Esercizio 2 (8 punti)

Sia m una funzione binaria che, dato un indice di macchina di Turing (MT) y e un ingresso x , restituisce il numero di mosse che impiega tale MT a terminare l'esecuzione su tale ingresso; m è indefinita qualora la macchina non termini.

1. La funzione m è calcolabile?
2. È decidibile il problema di stabilire se, data una generica MT con indice y , esista un ingresso x tale per cui
 - a. $m(y,x) = 0$?
 - b. $m(y,x) \leq y$?
 - c. $m(y,x) \neq \perp$?

Spiegare brevemente ma *esaurientemente e con chiarezza* le risposte.

Esercizio 3 (6 punti)

Si consideri il linguaggio $L_c = \{x.y \text{ in } \{0,1,2\}^* \mid \text{con } |x| = |y|, \text{ e tali che } x \text{ e } y \text{ coincidano per almeno 1 carattere nella stessa posizione}\}$.

Si descrivano una MT e una RAM che accettano il linguaggio L_c , valutandone le complessità temporali e spaziali, nel caso della RAM con entrambi i criteri di costo.

Esercizio 4 (9 punti)

- a) Si descriva un algoritmo che, ricevendo in ingresso un BST (binary search tree) e una coppia di chiavi appartenenti all'universo delle possibili chiavi usate nel BST, stampi l'elenco di tutte le chiavi presenti nell'albero il cui valore sia compreso tra le due chiavi fornite in ingresso. Si valuti la complessità asintotica dell'algoritmo fornito.
- b) E' possibile modificare l'algoritmo suddetto o costruirne uno diverso in modo da ottenere una migliore complessità, nel caso l'albero in ingresso sia un RB tree (albero rosso-nero)? Motivare brevemente la risposta.
- c) E' possibile modificare l'algoritmo suddetto o costruirne uno diverso in modo da ottenere una migliore complessità asintotica, nel caso l'albero in ingresso sia un RB tree e la coppia di chiavi individui un intervallo di possibili valori molto piccolo rispetto alla dimensione dell'albero (ad esempio decine di chiavi per alberi che contengano migliaia di nodi)? Motivare brevemente la risposta.

Tracce delle soluzioni

Esercizio 1

$$\forall n \forall m (f(n)=m \leftrightarrow n=0 \wedge m=0 \vee (n>0 \wedge m = n + f(n-1)))$$

Esercizio 2

Chiaramente m è calcolabile: è possibile infatti implementare un programma che emula l'esecuzione della y -esima MT contando il numero di passi eseguiti. A fine esecuzione viene restituito tale numero. Se invece l'esecuzione della macchina emulata non termina, allora neanche il programma termina, compatibilmente con la definizione di m .

Il punto 2a è decidibile, in quanto l'unico modo per terminare in 0 mosse è che lo stato iniziale della MT sia anche finale. Questo si può vedere semplicemente per ispezione della funzione di transizione della MT, che è ricostruibile mediante l'enumerazione di Gödel. Non è invece applicabile il teorema di Rice, poiché non si tratta di una proprietà della funzione calcolata dalla MT.

Anche il punto 2b è decidibile. Benché generico, y è dato e sappiamo che in al più y passi la MT può leggere al più y caratteri dell'ingresso. Il numero di stringhe diverse di al più y caratteri è finito; per ciascuna di esse basta verificare se la MT si arresta entro y passi.

Il punto 2c è indecidibile per il teorema di Rice (insieme delle funzioni definite in almeno un punto).

Esercizio 3

La MT copia in memoria la prima metà della stringa di ingresso, dopo averla individuata con una prima passata; indi confronta carattere per carattere la parte x memorizzata sul nastro con la parte y sul nastro di ingresso; al primo carattere coincidente si ferma accettando; rifiuta se non ne trova. La complessità spaziale e temporale è $\Theta(n)$.

La RAM si comporta sostanzialmente allo stesso modo, con la differenza di dover copiare l'intera stringa non essendo in grado di ritornare indietro sul nastro di ingresso. La verifica di coincidenza tra le due porzioni x e y può essere fatta confrontando due "puntatori" che procedono in parallelo. Le complessità spaziale e temporale a costo costante sono $\Theta(n)$, mentre a criterio di costo logaritmico sono rispettivamente $\Theta(n)$ e $\Theta(n \cdot \log(n))$.

Esercizio 4

- Un semplice algoritmo linearizza l'albero ottenendone un array ordinato. Indi scandisce linearmente l'array stampando le chiavi comprese nell'intervallo $[k_{\min} - k_{\max}]$. La complessità è $O(n)$, né è possibile ottenerne una più bassa, dato che, nel caso pessimo, potrebbe essere necessario stampare tutte le chiavi.
- Il fatto di usare un RB tree invece di un BST non cambia sostanzialmente le cose perché resta comunque necessario visitare tutto l'albero nel caso pessimo.
- In questo caso conviene usare un algoritmo diverso che cerca nell'albero tutte le chiavi dell'intervallo: siccome ogni singola ricerca costa $O(\log(n))$, l'esecuzione completa ha complessità $O(K \cdot \log(n))$, dove $K = k_{\max} - k_{\min}$, ossia un valore molto piccolo rispetto a n , tale da poterlo considerare costante.