

008AA – ALGORITMICA E LABORATORIO
Appello del 11 Gennaio 2010

Cognome Nome:

N. Matricola:

Corso: A B

Esercizio 1. (*4+2 punti*) Sia dato un albero binario T i cui n nodi contengono delle chiavi intere (positive o negative). Si vuole stabilire se esiste un nodo di T tale per cui la somma delle chiavi nel sottoalbero sinistro è uguale alla somma delle chiavi nel sottoalbero destro.

1. Progettare un algoritmo che risolve il problema suddetto. (*Si prega di accompagnare lo pseudo-codice con una descrizione informale dell'idea algoritmica sottostante.*)
2. Analizzare la complessità in tempo dell'algoritmo proposto.

Cognome Nome:

N.Matr:

Esercizio 2. (~~4~~+4 punti) Si consideri la sequenza di chiavi $S = (8, 3, 9, 4, 2, 16, 5)$.

1. Si progetti una tabella hash in cui le collisioni sono gestite con liste di trabocco e la sua dimensione m è tale per cui il *numero medio di chiavi* confrontate durante una ricerca sia almeno 2 e al più 3. Si inseriscano le chiavi di S in essa.
2. Si progetti una tabella hash di dimensione $m = 13$ in cui le collisioni sono gestite con indirizzamento aperto e hash doppio, e si inseriscano le chiavi di S in essa.

Cognome Nome:

N.Matr:

Esercizio 3. (*3+5+2 punti*) Dato un insieme di n oggetti aventi taglia $\{s_1, s_2, \dots, s_n\}$, descrivere un algoritmo per determinare se esiste un valore k tale che la somma totale delle taglie $s_i \leq k$ sia uguale alla somma totale delle taglie $s_i > k$.

1. Progettare un algoritmo di costo in tempo $O(n^2)$ al caso pessimo.
2. Progettare un algoritmo di costo in tempo $O(n \log n)$ al caso pessimo.
3. Nell'ipotesi che $\max_{i=1, \dots, n} s_i = O(n)$, progettare un algoritmo che richiede tempo ottimo $O(n)$ al caso pessimo.

Cognome Nome:

N.Matr:

Esercizio 4. (*4+2 punti*) Enunciare il Teorema Master (o principale) e poi utilizzarlo per la risoluzione della seguente relazione di ricorrenza: $T(n) = T(n/3) + O(n)$, se $n > 1$, e $T(1) = 5$.