ESAME DI ALGORITMI
Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
1 luglio 2025

Si risolvano i seguenti esercizi in un tempo non superiore a 3 ore. Si abbia cura di consegnare la risoluzione dei primi 3 esercizi in un foglio (Foglio A) separato da quello utilizzato per la consegna degli ultimi 3 esercizi (Foglio B).

——— Foglio A ———

1. In un albero rosso-nero, l'altezza nera di un nodo x, denotata $bh(x)$, è definita come il numero di nodi neri presenti lungo un qualunque cammino da x a una foglia `NIL`, escludendo il nodo stesso, ma includendo il nodo `NIL` finale (che si assume sempre nero). Per definizione degli alberi rosso-neri, tutti i cammini da un nodo verso le foglie devono avere la stessa altezza nera. Scrivere lo pseudo-codice di una procedura ricorsiva che, dato un nodo x, restituisca la sua altezza nera.

2. Si consideri un flusso di numeri interi che arrivano uno alla volta. Si desidera mantenere in ogni momento l'informazione sul k-esimo elemento più piccolo tra quelli finora ricevuti. Progettare e implementare una struttura dati che supporti le operazioni `add`(x), che inserisce un nuovo elemento intero x nel flusso, e `get`$()$, che restituisce il k-esimo elemento più piccolo tra quelli inseriti fino a quel momento. Se sono stati inseriti meno di k elementi, restituisce *non disponibile*.
   Suggerimento: si utilizzi una coda con priorità.

3. Scrivere lo pseudo-codice di una funzione ricorsiva, `verifica`(x, S) che, dato un nodo radice x di un albero rosso-nero e un intero S, verifichi se esiste almeno un cammino dalla radice x a una foglia tale che la somma delle chiavi dei nodi lungo il cammino sia esattamente S.

——— Foglio B ———

4. Si risolva l'equazione di ricorrenza $T(n) = 9T\left(\frac{n}{b}\right) + n^2 \log n$, al variare del parametro reale $b > 1$ utilizzando il metodo Master. Si stabilisca inoltre quale delle seguenti condizione sono soddisfatte dalla soluzione $T(n)$:
   (i) $T(n) = \Omega(n^2)$;
   (ii) $T(n) = \Theta(n \log n)$;
   (iii) $T(n) = o(n^2 \log^2 n)$.

5. Si definisca la proprietà di scelta greedy di un problema e si dimostri che Fractional Knapsack gode di tale proprietà.

6. Si scriva la formula ricorsiva utilizzata dall'algoritmo All-Pairs-Shortest-Paths basato sulla moltiplicazione di matrici e si simuli tale algoritmo per trovare la tabella (matrice) dei cammini minimi tra tutte le coppie di vertici del grafo in figura di cui si richiede la matrice di adiacenza.