“ALGORITMI E COMPLESSITÀ”
CORSO DI LAUREA MAGISTRALE IN INFORMATICA
UNIVERSITÀ DEGLI STUDI DI CATANIA
ANNO ACCADEMICO 2020/21

Terza sessione di esami – Secondo appello – 23 settembre 2021

Si svolgano i seguenti esercizi, argomentando adeguatamente le risposte.

ESERCIZIO 1

Utilizzando i tre metodi dell'analisi ammortizzata, si determini il costo ammortizzato per operazione di una sequenza di n operazioni, ove il costo c_i dell'i-esima operazione sia dato da

$$c_i = \begin{cases} 3 \cdot i & \text{se } i \text{ è potenza esatta di 4} \\ 5 & \text{altrimenti}\,. \end{cases}$$

ESERCIZIO 2

(a) Si definisca in maniera precisa la struttura dati B-tree e se ne illustri sinteticamente un'applicazione.

(b) Si effettui l'inserimento delle chiavi D, G, R, F, M, H, P, Q, I, L, A, B, C (nell'ordine dato) in un B-tree di grado minimo 2, inizialmente vuoto, e quindi si cancellino le chiavi F, L e D.

(c) Si forniscano una minorazione ed una maggiorazione dell'altezza h di un B-tree di grado minimo 2 in funzione del numero $n \geq 1$ di chiavi in esso contenute.

ESERCIZIO 3

Sia $G = (V, E)$ un grafo orientato con una funzione peso a valori reali positivi $w : E \to \mathbb{R}^+$, ma senza cicli di peso negativo. Siano inoltre a, b, c tre nodi distinti di G.
Si progetti un algoritmo efficiente, valutandone anche la complessità computazionale, per determinare (qualora esista) un ciclo di peso minimo (non necessariamente semplice) passante per i tre nodi a, b, c, in un ordine qualsiasi.

ESERCIZIO 4

(a) Si definiscano gli alberi binomiali, enunciando e dimostrando le loro più importanti proprietà.

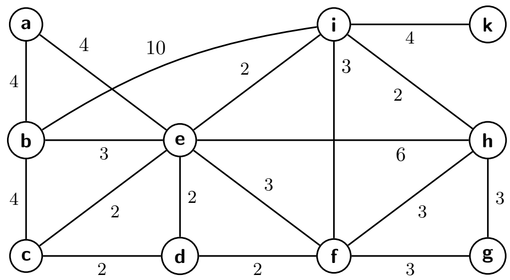
(b) Si definiscano gli heap binomiali, esibendo un esempio di heap binomiale con 29 chiavi.

(c) Si indichino le operazioni supportate dagli heap binomiali e con quale complessità.

(d) Si dimostri che un heap binomiale con n elementi è costituito da al più $\big(\lfloor \log_2 n \rfloor + 1\big)$ alberi binomiali.

ESERCIZIO 5

(a) Si descriva l'algoritmo di Kruskal, fornendone anche lo pseudo-codice, e lo si applichi al grafo a lato, utilizzando tra archi del medesimo peso l'ordinamento lessicografico (per cui, ad es., (a, b) precede (a, e) che, a sua volta, precede (i, k)), ove gli archi stessi sono rappresentati lessicograficamente (per cui, ad es., la rappresentazione di riferimento dell'arco tra i nodi e ed a è (a, e)).



(b) Si descrivano i “passi blu” e “passi rossi” negli algoritmi per il calcolo del *minimum spanning tree* e si enunci l'*invariante del colore*.
Quindi si dimostri che se dopo un certo numero di passi di colorazione il sottografo degli archi blu non forma ancora uno spanning tree, allora è possibile eseguire un passo blu.