SIMULAZIONE ESAME DI ALGORITMI

Università degli Studi di Catania
Tutorato di algoritmi e laboratorio 2023-2024
Corso di Laurea Triennale in Informatica
22 Marzo 2024

Foglio A

Esercizio 1

Dato il seguente array $A = \langle 2, 5, 4, 7, 9, 8, 10, 15, 11, 12, 17, 14, 16 \rangle$, svolgere i seguenti quesiti:

(a) Considerando un Max-Heap inizialmente vuoto, si inseriscano le chiavi seguendo l'ordine presentato nell'array A. Fornire la configurazione dell'albero Max-Heap dopo ciascuna delle 13 operazioni di inserimento. Denominare l'heap risultante H_1.

(b) Creare un nuovo heap H_2 utilizzando l'array iniziale A senza le operazioni effettuare al punto (a) e utilizzando la procedura Build-Min-Heap invece degli inserimenti, ottenendo così un Min-Heap al posto di un Max-Heap.

(c) Indicare quale dei due heap, H_1 o H_2, sarebbe più adeguato per ottenere un ordinamento non crescente dell'array A mediante l'algoritmo di ordinamento Heapsort. Successivamente, illustrare i passi dell'ordinamento Heapsort, fornendo sia la configurazione dell'albero heap sia lo stato dell'array A, comprese le chiavi già estratte.

Esercizio 2

Si consideri un albero Rosso-Nero completo, contenente 15 chiavi. I nodi dell'albero sono colorati di nero con l'eccezione di quelli situati al livello 1 e al livello 3, che sono di colore rosso. Si eseguano quattro operazioni di inserimento di una chiave ciascuna, assicurandosi che ogni nuova chiave inserita sia maggiore di tutte quelle precedentemente presenti nell'albero. Successivamente, si eseguano quattro operazioni di cancellazione, rimuovendo ad ogni passo la chiave più piccola presente nell'albero. Infine, si rimuova la radice dell'albero e si fornisca la configurazione finale risultante.

Esercizio 3

Sia data la sequenza di 6 matrici definito dal vettore delle dimensioni $\langle 4, 2, 2, 5, 3, 3, 2 \rangle$ e si supponga di eseguire l'algoritmo di programmazione dinamica per il calcolo della parentesizzazione ottima di una sequenza di matrici. Si fornisca il costo ottimale della moltiplicazione della sequenza fornita in input, la matrice ottenuta alla fine dell'esecuzione dell'algoritmo e la relativa parentesizzazione ottima.

Esercizio 4

Si forniscano lo pseudo-codice o il codice nel linguaggio C/C++ dell'algoritmo di Bellman-Ford per il calcolo dei cammini minimi da una singola sorgente in un grafo diretto e pesato e si illustri la complessità temporale dell'algoritmo. Dati un grafo pesato (G,w), con G = (V, E) e $w : E \to \mathbb{R}$, e due vertici s_1, $s_2 \in$ V, si illustri un algoritmo efficente (mediante pseudo-codice e valutandone la complessità computazionale) per determinare se il grafo (G,w) contenga cicli di peso negativi raggiungibili da s_1 oppure s_2

Foglio B

Esercizio 5

Si consideri l'equazione di ricorrenza $T(n) = 3T\left(\frac{n}{b}\right) + \Theta(nlogn)$.

1. Si risolva l'equazione al variare del parametro reale b $\geq$
2. Si stabilisca per quali valori di b la soluzione T(n) all'equazione soddifsa le seguenti condizioni:
   (a) $T(n) = \mathcal{O}(n\log^2 n)$
   (b) $T(n) = \Theta(n)$
   (c) $T(n) = \omega(n \log n)$

Esercizio 6

In un grafo non pesato, la lunghezza di un cammino tra due vertici è definita come il numero di archi che compongono il cammino. Si consideri il problema Unweighted Shortest Path, in cui l'input è un grafo G = (V, E) orientato e non pesato, e due vertici distinti u, v $\in$ V . L'obiettivo del problema è trovare un cammino di lunghezza minima. Si dimostri che Unweighted Shortest Path ha la propriet di sottostruttura ottima