

Esercizi di Strutture Dati Algoritmi e Complessità

Luca Becchetti

May 20, 2024

1 Esercizi del 20 Maggio 2024

Ridimensionamento di una tabella hash dinamica [1, Capitolo 11 e Sezione 16.4]

Si consideri una tabella hash dinamica realizzata con array e nella quale le collisioni sono gestite con liste di trabocco: quando il fattore di carico della tabella supera il valore $1/2$ si copia il contenuto della tabella in una nuova tabella avente dimensione *doppia* rispetto alla precedente. Poiché la tabella è realizzata con un array, occorre riassegnare ciascun elemento presente al momento del raddoppio a una posizione del nuovo array usando una nuova funzione hash,¹ dunque con un costo proporzionale al numero di elementi presenti nella tabella al momento del raddoppio.

Quesito. Si supponga che la dimensione iniziale della tabella sia pari a 1. Si consideri una successione di n inserimenti in una tabella hash dinamica gestita come descritto sopra, supponendo che queste siano le uniche operazioni (quindi non vi sono cancellazioni). Dimostrare che il costo atteso per inserimento è $\mathcal{O}(1)$, supponendo che le funzioni hash usate si comportino in modo ideale, ossia soddisfino le ipotesi di uniformità e indipendenza.

Heap incredibili

Il professor Knowitall sostiene di aver concepito una nuova versione degli heap (la chiameremo K-Heap) che consente le seguenti operazioni su chiavi appartenenti a un universo perfettamente ordinato. 1) INSERT(x): inserisce la chiave x nella struttura dati; 2) EXTRACTMIN(): estrae e restituisce la chiave minima presente nell'heap modificato. Il Prof. Knowitall sostiene inoltre che le operazioni descritte sopra sono basate *esclusivamente* su confronti tra le chiavi e hanno costo $\mathcal{O}(1)$ nel caso peggiore.

Quesito. Dimostrare in modo rigoroso che il Prof. Knowitall ha preso un abbaglio.

Cammini minimi

Si consideri un grafo non orientato $G = (V, E)$, non necessariamente connesso, in cui tutti gli archi hanno lo stesso peso positivo $\alpha > 0$.

Quesito. 1) Scrivere lo pseudo-codice di un algoritmo efficiente che, dato in input il grafo G di cui sopra e due nodi u e v , calcola la lunghezza del cammino minimo che collega u e v . Se i nodi non sono connessi, la distanza è infinita. Non è consentito usare variabili globali. 2) Analizzare il costo computazionale dell'algoritmo proposto al punto 1). **Nota.** Se come subroutine si

¹La funzione hash cambia perché è cambiata la dimensione della tabella.

usano algoritmi noti visti a lezioni, non è necessario scriverne il codice ma soltanto dichiarare esplicitamente quale algoritmo si usa, identificare chiaramente il nome della relativa subroutine e gli argomenti di quest'ultima.

Visita in profondità

Si consideri il grafo orientato in Figura 1. Si consideri la DFS su tale grafo, supponendo che essa esamini i vertici in ordine alfabetico e che ogni lista di adiacenza sia ordinata alfabeticamente.

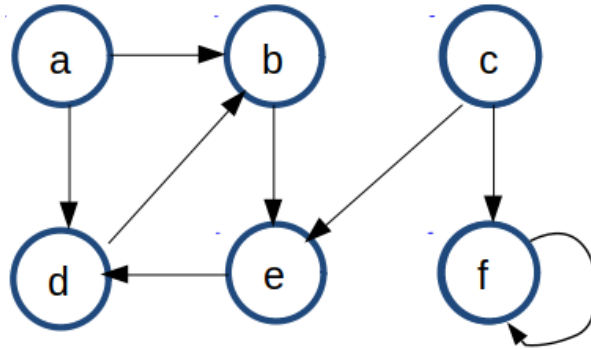


Figure 1: Grafo diretto.

Quesito. 1) Illustrare la struttura a parentesi della visita in profondità del grafo in figura; 2) indicare la classificazione di ciascun arco (si usi T per denotare gli archi d'albero B per gli archi all'indietro, F per gli archi in avanti e C per gli archi trasversali).

References

- [1] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduzione agli algoritmi e strutture dati* 4/ed. 2023.