LAB. DI ALGORITMI

APPUNTI A CURA DI: RICCARDO LO IACONO

Università degli studi di Palermo
a.a. 2023-2024

Indice.

– 1 – Strutture dati astratte: alberi.

Tra le varie strutture dati astratte, gli alberi sono sicuramente quelli maggiormente utilizzati e di maggior importanza. Di questi, ne esistono molteplici varianti, ciascuna con uno scopo ben preciso; fatto sta che tutte queste varianti sono accomunate dall'efficienza.

– 1.1 – BST: binary search trees.

Prima di procedere con il discutere varianti di alberi più complesse, si procede a fare un richiamo al concetto di albero binario di ricerca. Questi si ricorda essere una tipologia di albero binario che, dato S un insieme di elementi ordinati, memorizza gli stessi in un nodo dell'albero in modo tale che, posto $x \in S$, si abbia

- per ogni altro y nel sotto-albero sinistro con radice x, si abbia

  $$key[y] \leq key[x]$$

  cioè, ogni elemento del sotto-albero sinistro deve avere un valore minore o uguale, a quello della radice del sotto-albero stesso;

- per ogni ogni altro y nel sotto-albero destro radicato in x, si abbia

  $$key[y] > key[x]$$

  cioè, ogni elemento del sotto-albero sinistro deve avere un valore maggiore, a quello della radice del sotto-albero stesso.

Si ricorda brevemente che, posto h l'altezza dell'albero, le operazioni di inserimento, ricerca e cancellazione sono tutto di costo $\mathcal{O}(h)$. Si ha quindi che, poiché

$$h = \begin{cases} \log_2(n) \text{ , se l'albero è perfettamente bilanciato;} \\ \log_2(n) \leq k \leq n \text{ , se l'albero non è perfettamente bilanciato;} \\ n \text{ , se completamente sbilanciato,} \end{cases}$$

nel caso pessimo si ha un costo di $\mathcal{O}(n)$.

– 1.2 – AVL trees: Adelson-Velsky-Landis trees.

Gli AVL sono una tipologia di alberi binari di ricerca bilanciati in altezza. Nello specifico, si dice che un AVL è bilanciato se questi ha, per ogni sotto-albero, un *fattore di bilanciamento* B_f minore o uguale ad uno.

Qui per fattore di bilanciamento si intende la differenza in modulo tra l'altezza dei due sotto-alberi.

Per quel che riguarda le operazioni: essendo, come detto, che gli AVL sono dei BST, e poiché essa non modifica la struttura dell'albero, la ricerca è analoga a quella dei BST; inserimento e cancellazione viceversa, proprio perché modificano la struttura dell'albero, e rischiano di sbilanciarlo, sono modificate in modo tale che a seguito di esse l'albero risulti ancora bilanciato. Tale modifica consiste nelle operazioni di rotazione descritte a seguito.

Osservazione. Sebbene l'aggiunta delle operazioni di rotazione nel caso di inserimento e cancellazione, ciascuna delle tre operazioni richiede al più tempo $\mathcal{O}(\log_2(n))$: questo perché proporzionali all'altezza dell'albero, e poiché il costo di ogni rotazione è $\mathcal{O}(1)$.

– 1.2.1 – Ribilanciamento di un AVL.

Alla base del processo di bilanciamento vi sono le operazioni di rotazione a sinistra e a destra. Per comprendere tali operazioni, si faccia riferimento a *Figura 1.a.* Si supponga di aggiungere ad u un nodo, portando così a uno sbilanciamento dell'albero. In questo caso si dimostra sufficiente una singola rotazione a destra, a seguito della quale l'albero risulta bilanciato. L'operazione appena descritta prende il nome di rotazione destra-destra, a questa si aggiungono la rotazione sinistra-sinistra (simmetrica alla rotazione destra-destra) e le rotazioni sinistra-destra, destra-sinistra tra loro simmetriche.

1.a 1.b

Osservazione. Nel caso di rotazioni sinistra-sinistra (destra-destra equivalentemente) si dimostra sufficiente un'unica operazioni di rotazione elementare.

Sia ora considerato il caso di una rotazione sinistra-destra (nel caso destra-sinistra si procede invertendo l'ordine delle operazioni), per farlo si faccia riferimento a figura *Figura 2.a*. Si supponga che ad y si aggiunga un nodo, sbilanciando in tal modo l'albero. Come mostrato in *Figura 2.b*, una sola rotazione a sinistra non è sufficiente a bilanciare l'albero; a questa è infatti necessario aggiungere una rotazione a destra

2.a 2.b

2.c 2.d

Figura 2.c, portando ad un AVL bilanciato come mostrato in *Figura 2.d.*