

Alberi, alberi binari

Lezione 22 di Programmazione 2

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)

Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Studi in Informatica applicata, AA 2007-8

Indice

1. Alberi, alberi binari
2. alberi: definizioni
3. alberi: terminologia
4. rappresentazione di alberi binari
5. rappresentazione binaria di foreste
6. alberi liberi e grafi
7. alberi binari: proprietà matematiche (1)
8. alberi binari: proprietà matematiche (2)
9. esercizi

alberi: definizioni

“popolarità” degli alberi in informatica:

strutture dati di frequente impiego negli algoritmi, v. ad es. quelli studiati per il problema della connettività

le dinamiche di algoritmi hanno spesso struttura ad albero, ad es. invocazioni ricorsive in un algoritmo divide et impera

gli alberi sono inoltre presenti in una miriade di applicazioni: alberi genealogici, organigrammi, strutture di file system, strutture di libri, alberi sintattici, termini, ...

cos'è, esattamente, un albero? ecco un paio di definizioni, per cominciare:

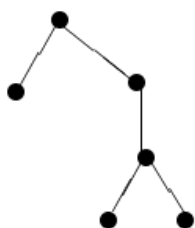
albero libero: grafo connesso privo di cicli

N.B. equivalentemente: grafo in cui esiste un unico cammino fra ogni coppia di vertici

albero (radicato): albero con un vertice designato, detto **radice**

N.B. sebbene il grafo di un albero sia generalmente non orientato, la designazione di un vertice radice induce naturalmente (per induzione) un ordinamento parziale dei vertici, grazie all'assenza di cicli, dunque un orientamento degli archi

alberi: terminologia



cammino:

sequenza di vertici distinti, con vertici successivi connessi da archi

ordinamento dei vertici:

$m < n$ se m precede n nel cammino dalla radice a n

terminologia: **padre**, **figli**, **foglie**, **sottoalbero**, ...

un albero è **k-ario** se ogni vertice ha al più al più k figli

un albero **k-ario** è **completo** se ogni vertice interno (cioè non foglia) ha k figli

un albero si dice **ordinato** se è definito un ordine (totale) sui figli di ciascun padre
i termini hanno struttura ad albero ordinato

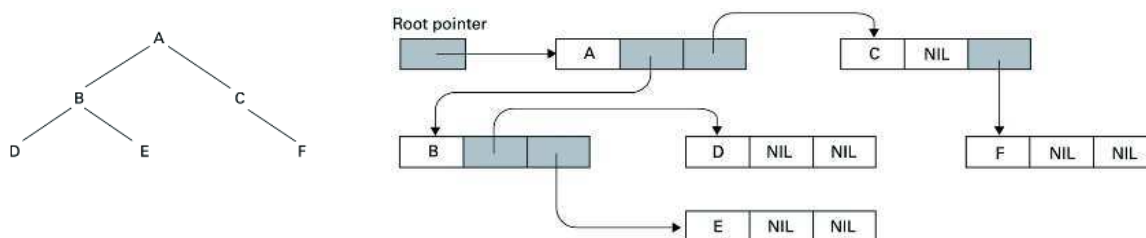
l'**altezza** di un albero è la massima lunghezza (in n. di archi) dei cammini dalla radice
è detta anche **profondità** dell'albero

racpresentazione di alberi binari

albero **binario** = albero **k-ario**, con $k = 2$

si può rappresentare ciascun vertice con una struttura elementare dotata di **due puntatori** (che lo collegano ai due figli, se entrambi presenti, altrimenti uno dei puntatori o entrambi sono nulli)

si rappresentano in tal modo alberi binari **ordinati**



```
struct btree { Elem elem; btree *l, *r; };
typedef btree *btlink;
```

racpresentazione binaria di foreste

la rappresentazione vista degli alberi binari ben si presta ad operazioni su di essi per l'esecuzione delle quali si proceda dalla radice verso le foglie

altrimenti, si può estendere la struttura con un terzo link, che punta al vertice padre

la suddetta rappresentazione si generalizza facilmente ad alberi k-ari, dotando di k link la struttura di rappresentazione dei vertici, oltre all'eventuale link al vertice padre, quando ciò convenga

in tutti questi casi, si può anche scegliere una rappresentazione basata su array e indici invece di strutture e puntatori, poiché è prefissato un limite massimo al numero dei figli di qualsiasi vertice

come rappresentare alberi in cui tale limite non c'è?

una soluzione consiste nel rappresentare l'insieme (ordinato o meno) dei figli di un vertice interno mediante una **lista**

questa soluzione riporta ad una struttura con due soli link, come nel caso degli alberi binari...

qui, però, un link punta al figlio più a sinistra, l'altro punta a un **fratello** del vertice

questa rappresentazione stabilisce una corrispondenza biunivoca fra alberi binari ordinati e **foreste** ordinate, v. esercizio 1

una **foresta** è un insieme di alberi che non condividono vertici

una foresta è **ordinata** se tale insieme è totalmente ordinato e gli alberi sono ordinati

alberi liberi e grafi

gli alberi liberi sono grafi (non orientati) che soddisfano una **proprietà caratteristica**; questa può essere enunciata in vari modi: un grafo con N vertici è un albero libero se soddisfa una delle seguenti **condizioni equivalenti**:

- è connesso e privo di cicli

- ha $N-1$ archi ed è privo di cicli

- ha $N-1$ archi ed è connesso

- esiste esattamente un cammino (semplice) fra ogni coppia di vertici distinti

- è minimamente connesso, cioè è connesso, ma l'eliminazione di un qualsiasi arco lo rende non connesso

un cammino in un grafo è una sequenza di vertici in cui vertici successivi sono connessi da un arco: il cammino è **semplice** se ogni vertice in esso occorre una sola volta

una **foresta libera** è un grafo in cui ogni cammino è semplice

ogni cammino è semplice anche in un albero libero, caso particolare di foresta libera

analogamente a quanto visto per gli alberi liberi, è possibile fornire caratterizzazioni equivalenti del concetto di foresta libera, v. esercizio 2

alberi binari: proprietà matematiche (1)

un albero binario completo con N vertici interni ha $N+1$ foglie

per induzione sull'altezza dell'albero: la proprietà vale per l'albero di un solo vertice, e per alberi con più vertici vale sui sottoalberi della radice per ipotesi d'induzione

un albero binario completo con N vertici interni ha $2N$ archi: $N-1$ fra vertici interni e $N+1$ con foglie

segue dalla proprietà precedente e dall'unicità dell'arco fra ogni vertice (non radice) e il padre

definizioni:

livello di un vertice in un albero: numero di vertici che lo precedono nel cammino dalla radice

altezza di un albero: massimo dei livelli dei suoi vertici

lunghezza del cammino di un albero: somma dei livelli dei suoi vertici

lunghezza del cammino interno di un albero: somma dei livelli dei suoi vertici interni

lunghezza del cammino esterno di un albero: somma dei livelli delle sue foglie

in un albero binario completo con N vertici interni, se I è la lunghezza del cammino interno ed E è la lunghezza del cammino esterno, allora $E = I + 2N$

qualsiasi albero binario completo con N vertici può essere costruito partendo dall'albero che consta della sola radice e iterando quindi N volte il rimpiazzamento di una foglia con un vertice interno padre di due foglie: se k è il livello di una foglia rimpiazzata, la lunghezza del cammino interno aumenta di k mentre quella del cammino esterno aumenta di $k+2$ (due nuove foglie di livello $k+1$ meno una di livello k)

alberi binari: proprietà matematiche (2)

l'altezza h di un albero binario completo con N vertici interni soddisfa:

$$\lfloor \lg N \rfloor + 1 \leq h \leq N-1$$

il limite inferiore si ha con alberi **bilanciati** (v. esercizio 6)

cioè alberi binari in cui le altezze dei due sottoalberi di qualsiasi vertice interno differiscono di 1 al più

quello superiore quando i vertici interni formano una **catena**

la lunghezza del cammino interno, I , di un albero binario completo con N vertici interni soddisfa: $N \lg(N/4) < I \leq N(N-1)/2$

i limiti inferiore e superiore si hanno con gli stessi alberi di cui sopra

per il limite inferiore, si noti che un albero completo bilanciato ha $N+1$

foglie di livello almeno $\lfloor \lg N \rfloor$, di cui almeno due di livello $\lfloor \lg N \rfloor + 1$:

moltiplicando e applicando l'ultima proprietà dalla pagina precedente si ha

$$I = E - 2N > (N+1) \lfloor \lg N \rfloor - 2N > N \lg(N/4) \text{ (v. esercizio 7)}$$

gli alberi bilanciati sono spesso utili al progetto di algoritmi **ottimali**

esercizi

1. dimostrare che esiste una biiezione fra alberi binari ordinati e foreste ordinate
2. generalizzare la caratterizzazione degli alberi liberi qui data da cinque condizioni equivalenti sui grafi, ad una analoga caratterizzazione delle foreste libere
3. su N vertici si possono costruire N^{N-2} alberi liberi: quanti alberi radicati?
quanto asserito è stato dimostrato da Cayley nell'800; a Prüfer (1918) si deve un'elegante dimostrazione algoritmica:
<http://www.mathworld.wolfram.com/PrueferCode.html> (v. anche nota EWD677 di E.W. Dijkstra)
4. due alberi (ordinati o meno) sono **isomorfi** se esiste una biiezione fra i vertici che rispetta la relazione binaria costituita dagli archi
cioè: se φ è una tale biiezione, ε_1 la relazione di esistenza di arco nel primo albero, ε_2 quella nel secondo albero, allora per ogni coppia di vertici u, v nel primo albero: $u \varepsilon_1 v \Leftrightarrow \varphi(u) \varepsilon_2 \varphi(v)$
dimostrare che se due alberi ordinati rappresentano lo stesso albero non ordinato allora sono isomorfi, ma che non vale l'implicazione inversa
5. progettare e implementare un algoritmo per stabilire se due alberi ordinati, rappresentati da alberi binari ordinati come indicato qui, rappresentino lo stesso albero non ordinato, e valutarne le prestazioni sia in via analitica che empirica
6. dimostrare che, diversamente dall'enunciato della Proprietà 5.8 nel testo (Sedgewick, 2003) a p. 243, l'altezza di un albero binario completo bilanciato di N vertici interni è almeno $\lfloor \lg N \rfloor + 1$
7. dimostrare che la lunghezza del cammino interno di un albero binario completo bilanciato di N vertici interni è maggiore di $N \lg(N/4)$