Liste multiple e rappresentazione di grafi

Lezione 12 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 2006-7

Indice

liste multiple

una **lista multipla**, o **multilista**, è una lista concatenata la cui struttura dei nodi ha **due o più link** ad altri nodi

esempio già visto: le **liste doppiamente concatenate**

le multiliste sono utili per strutture dati con **più ordinamenti** simultaneamente

ad esempio: una lista di elementi dei quali si voglia rappresentare sia l'ordine cronologico di inserimento nella lista sia un altro ordine, ad es. basato su un ordinamento definito sul tipo degli elementi

è discutibile se, e in qual modo, le multiliste siano utili a rappresentare **matrici multidimensionali sparse:**

una matrice di elementi di un tipo in cui sia definito un **valore nullo** è detta **sparsa** se quasi tutti i suoi elementi sono nulli; nel caso opposto è detta **densa**
è evidente il risparmio di spazio nel rappresentare matrici sparse mediante **liste**, con un nodo per ogni elemento non nullo (e nessuno per quelli nulli)
naturalmente, un simile risparmio si paga con un maggior tempo di accesso
non è però evidente la convenienza, in generale, di avere tanti link per nodo quante sono le dimensioni della matrice (v. esercizi)

grafi: matrici di adiacenza

i **grafi**, strutture matematiche già familiari dall'analisi del problema della connettività, hanno una immediata rappresentazione con **matrici di adiacenza**: matrici binarie, in cui l'elemento (i,j) è 1 o 0 a seconda che esista o meno un arco dal vertice i al vertice j

si assume di solito che la relazione binaria espressa dal grafo sia **riflessiva**, ossia che esista un arco (anche se non esplicitamente tracciato nel grafo) da ciascun vertice a sé stesso

i bit sulla diagonale della matrice di adiacenza hanno dunque valore 1, per convenzione

la rappresentazione con matrice di adiacenza ben si presta anche a **grafi orientati**: la matrice di adiacenza è **simmetrica** se il grafo **non** è orientato (relazione binaria simmetrica)

di solito, se il grafo ha **V** vertici ed **E** archi, se ne rappresentano i vertici con i primi **V** numeri naturali e gli archi con corrispondenti **E** coppie di tali numeri

se il grafo ha pochi archi, ossia $E << V^2$, allora la matrice di adiacenza è **sparsa**, nel qual caso rappresentazioni alternative del grafo meritano considerazione

non solo per il risparmio di spazio ma anche perché, per certi algoritmi, se ne ottiene un minor **tempo di esecuzione**

rappresentazione di grafi mediante liste

una rappresentazione dei grafi alternativa alla matrice di adiacenza si realizza mediante **liste di adiacenza:**

> si rappresenta un grafo di **V** vertici con un array di altrettanti puntatori a liste (unidimensionali): la lista di indice **v** contiene un nodo per ciascun vertice connesso da un arco al vertice **v** (l'ordine dei nodi in ciascuna lista è arbitrario)

con la rappresentazione mediante liste di adiacenza si perde, rispetto a quella della matrice di adiacenza, il vantaggio di poter determinare l'**esistenza di un arco**, fra due vertici dati, in **tempo costante**

tuttavia, non tutti gli algoritmi hanno in questo il fattore critico di prestazione in termini di tempo di esecuzione:

> ad esempio, per algoritmi che debbano frequentemente elaborare informazione relativa a **tutti gli archi** del grafo, una rappresentazione lineare in **V+E**, quale quella delle liste di adiacenza, può ben risultare vantaggiosa rispetto a quella della matrice di adiacenza, quadratica in **V**, se questa è sparsa

esercizi

il testo (Sedgewick, 2003), a p.121, suggerisce l'uso di una multilista per rappresentare una matrice multidimensionale sparsa, con un nodo per ogni elemento (non nullo) della matrice e "un link per ogni dimensione, dove un link punta al successivo nodo per quella dimensione"; questo suggerimento ha il difetto di presumere un unico ordinamento degli elementi della matrice per ciascuna dimensione:

1. mostrare che il risparmio di spazio suggerito dal testo è conseguibile mediante rappresentazione con una lista unidimensionale
2. in quanti modi può definirsi un ordinamento lessicografico degli elementi della matrice, determinato solo dai valori degli indici e basato su un ordinamento (totale) delle dimensioni, per una matrice N-dimensionale?
3. in quali casi l'uso di una multilista per la rappresentazione in questione può presentare vantaggi rispetto all'uso di una lista unidimensionale, e quali vantaggi?

esercizio: cammino Hamiltoniano

un cammino in un grafo è **Hamiltoniano** se ne attraversa tutti i vertici, attraversando ciascuno di essi una sola volta; il problema di decidere se un grafo possieda o meno un cammino Hamiltoniano è risolto costruttivamente quando l'algoritmo di risoluzione, in caso di risposta positiva, esibisce un tale cammino

1. analizzare le differenze di prestazione da attendersi nel tempo di esecuzione di tali algoritmi costruttivi, per grafi non orientati con matrice di incidenza sparsa, tra la rappresentazione del grafo come matrice di incidenza e quella mediante liste di incidenza
2. considerare la questione precedente per grafi orientati: cambia qualcosa?
3. progettare un algoritmo costruttivo per il problema di decisione relativo all'esistenza di un cammino Hamiltoniano in un grafo non orientato e codificarlo come funzione C++ in due modi, rispettivamente con la rappresentazione mediante matrice di incidenza e con quella mediante liste di incidenza; procedere quindi alla stesura di un programma client di tali funzioni per il confronto empirico delle rispettive prestazioni rispetto al tempo di esecuzione, nel caso di matrici di incidenza sparse, ed effettuare il confronto con grafi aventi `V = 10, 100, 1000, 10000`, ed `E = V, 2V, 3V, 4V` in ciascun caso
4. adattare gli algoritmi e relativo codice dell'esercizio precedente ai grafi orientati ed eseguire l'analogo confronto empirico delle prestazioni

esercizio: il problema KWIC

l'acronimo sta per **Key Word In Context**

problema proposto nel 1972 da D.L. Parnas quale caso di studio per il confronto di diversi approcci al progetto di algoritmi; il problema è anche d'interesse pratico: la sua soluzione è utile a realizzare efficienti algoritmi di indicizzazione, ad esempio per il comando Unix **man**

definizioni:

un **testo** è una sequenza di righe

una **riga** è una sequenza di parole

una **parola** è una sequenza di caratteri alfanumerici; sull'alfabeto di tali caratteri è definito un ordinamento totale, esteso lessicograficamente alle parole e alle righe

una **permutazione circolare di una riga** si ottiene spostando la sua prima parola all'ultima posizione e decrementando la posizione di ciascuna delle altre parole; iterando questa operazione si ottengono tutte le permutazioni circolari della riga

problema: dato un testo in input, produrre una rappresentazione dell'insieme lessicograficamente ordinato delle permutazioni circolari delle sue righe