

Programmazione dei Calcolatori con Laboratorio

Simulazione esame 4
Aprile, 2018

Consideriamo un insieme U composto dagli interi da 0 a $n-1$. Ad ogni elemento i di U vogliamo associare un sottoinsieme U_i di U stesso. Questo viene fatto attraverso un array a di n liste, una per ogni elemento in U . Ogni lista $a[i]$ contiene gli elementi di U_i in un ordine qualsiasi.

La matrice di adiacenza m degli insiemi U_i è un array di lunghezza $n \times n$ a valori in $\{0, 1\}$ tale che per i e j nell'intervallo $0, \dots, n-1$

$$m[n * i + j] \equiv 1 \text{ se e solo se } j \in U_i.$$

Si scriva una funzione C che dato un array a di n liste come descritto sopra, restituisce la matrice di adiacenza m degli n insiemi U_i descritti dalle liste in a .

I sottoinsiemi che compongono l'array in input sono rappresentati con liste concatenate i cui nodi sono definiti come segue:

```
struct nodo {
    int chiave;
    struct nodo *succ, *prec;
};
typedef struct nodo nodo;
```

Ogni nodo è collegato a quello che lo precede attraverso il puntatore `prec` ed a quello che lo segue per mezzo del puntatore `succ`. Il campo `prec` del primo nodo della lista ed il campo `succ` dell'ultimo valgono NULL.

La funzione da progettare deve avere il seguente prototipo:

```
int *Liste2MatriceAdiacenza( nodo *a[], int n );
```

dove a è l'array di n liste che rappresentano i sottoinsiemi, ovvero $a[i]$ è il puntatore al primo nodo della lista che rappresenta il sottoinsieme U_i , n è il numero di liste nell'array a . La funzione deve restituire la matrice di adiacenza con le proprietà descritte nel testo.

Dopo aver progettato ed implementato la funzione, se ne calcoli il costo sia in termini di tempo che di memoria supplementare utilizzata (al netto di quella utilizzata per memorizzare l'input e l'output).

Modalità di consegna: Lo studente deve consegnare **un unico** file denominato `CognomeNome.c` (dove `Cognome` e `Nome` stanno rispettivamente per il proprio cognome ed il proprio nome).

Tale file, con l'aggiunta di una opportuna funzione `main` che non deve essere contenuta nel file consegnato, deve poter essere compilato senza errori. Quindi deve contenere:

- la funzione richiesta (`Liste2MatriceAdiacenza()`) che a sua volta deve rispettare le specifiche imposte dal problema;
- tutte le definizioni di tipi e `struct`;
- ogni altra funzione utilizzata dalla soluzione in quanto non è permesso l'utilizzo di funzioni non definite all'interno del file consegnato;
- tutti gli *header* delle librerie utilizzate.

Infine la funzione `main()` **non** deve essere inclusa nel file `CognomeNome.c` pertanto si consiglia di definirla in un secondo file denominato `main.c`. I due file possono essere compilati insieme utilizzando il comando

```
gcc main.c CognomeNome.c
```

Inviare le soluzioni al seguente indirizzo email entro il 25 aprile 2018.

tutor.programmazionec@gmail.com