

03AAX Algoritmi e Strutture Dati

Appello del 20/09/2023 - Prova di programmazione (12 punti)

1. (2 punti)

Sia data una matrice M di dimensione $r \times c$ contenente elementi interi.

Scrivere una funzione che generi una matrice M' di dimensione $r \times c$ derivata da M in cui ogni elemento $[i][j]$ assume il valore della somma cumulata dei primi k elementi adiacenti lungo la riga i e la colonna j , in entrambe le direzioni, escluso l'elemento $[i][j]$ originale. Nel caso non ci fossero elementi sufficienti lungo una certa direzione, ci si limiti a usare quelli disponibili.

La matrice M' sia allocata dentro alla funzione.

Completare opportunamente il prototipo in modo che la nuova matrice sia disponibile al chiamante.

```
void f(int **M, int r, int c, int k, ...);
```

Esempio:

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & -2 & 5 \\ 1 & 2 & 3 & 4 \\ -2 & -3 & -1 & 0 \\ -5 & 1 & -2 & 9 \end{pmatrix} \xrightarrow{k=2} M' = \begin{pmatrix} 8 & 13 & 8 & 14 \\ 1 & 6 & 15 & 9 \\ 1 & 11 & 5 & 23 \\ -6 & 3 & -6 & 14 \\ -2 & 1 & 7 & 3 \end{pmatrix}$$

2. (4 punti)

Si fornisca la definizione delle strutture dati `LIST` e `NODE`, come ADT di I categoria e quasi ADT rispettivamente, per rappresentare una lista doppio linkata di caratteri, senza sentinelle. Suddividere il codice in modo opportuno tra file `.h` e `.c`.

Si scriva una funzione `void f(LIST l, int k)` che ricevuta in input una lista rappresentata facendo riferimento ai tipi definiti in precedenza, già ordinata in ordine alfabetico crescente, compatti i contenuti della lista cancellando tutti i nodi uguali consecutivi, ad eccezione del primo di ogni gruppo, solo se la sotto-lista di nodi uguali consecutivi è composta k da almeno elementi.

Esempio:

$$a \leftrightarrow a \leftrightarrow b \leftrightarrow b \leftrightarrow b \leftrightarrow c \leftrightarrow d \leftrightarrow d$$

diventa per $k=3$

$$a \leftrightarrow a \leftrightarrow b \leftrightarrow c \leftrightarrow d \leftrightarrow d$$

3. (6 punti)

Dato un multi-insieme di interi $S = \{a_1, \dots, a_n\}$ di dimensione nota n e un intero $0 < s \leq \sum_{i=1}^n a_i$ identificare, se possibile, due multi-insiemi disgiunti $S_1 \subset S$ e $S_2 \subset S$ tali per cui valga la relazione:

$$\sum_{j=1}^{|S_1|} a_j = s + \sum_{k=1}^{|S_2|} a_k$$

A fronte di più soluzioni ammissibili, preferire quella che usa più elementi possibili di S , ossia per cui si abbia $\max \{|S_1| + |S_2|\}$.

03AAX Algoritmi e Strutture Dati

Appello del 20/09/2023 - Prova di programmazione (18 punti)

Descrizione del problema

Dato un grafo non orientato e pesato $G = (V, E)$, si definisce *k-capacitated tree partition* una collezione E_1, \dots, E_n di sottoinsiemi di E *disgiunti per vertici*, tali per cui per ogni E_i il grafo indotto è un albero di almeno k vertici. Tutti i vertici del grafo originale devono appartenere a uno dei grafi indotti. Per gli scopi del problema si è interessati all'individuazione di un partizionamento tale per cui valga $\min \sum_i \sum_{e \in E_i} w(e)$, dove $w(e)$ rappresenta il peso del generico arco $e \in E_i$. Sia k un parametro del problema.

Richieste del problema

Strutture dati e letture

Scrivere la definizione e implementazione delle strutture dati repute necessarie a modellare le informazioni del problema, quali il grafo e il relativo tree partitioning, e le funzioni di acquisizione dei dati stessi. In caso di organizzazione delle strutture dati su più file, indicare esplicitamente il modulo di riferimento.

Si assuma che il grafo in input sia riportato in un file di nome `grafo.txt`, organizzato come segue:

- Sulla prima riga appare il numero V di vertici;
- Seguono un numero indefinito di righe riportante coppie (u, v) di interi, con $u \geq 0$ e $v < V$ a rappresentare gli archi del grafo, e il relativo peso $w_{(u,v)}$.

Problema di verifica

Data una soluzione proposta, verificare che questa rappresenti effettivamente un k -capacitated tree partitioning, tenendo in considerazione la definizione teorica proposta in precedenza. Si trascuri la richiesta di minimalità della somma complessiva dei pesi per gli archi scelti.

La soluzione proposta deve essere letta da file il cui nome e formato è a discrezione del candidato, che è tenuto a fornire anche una breve spiegazione dei contenuti del file stesso.

Problema di ricerca e ottimizzazione

Identificare, se possibile, un k -capacitated tree partitioning che rispetti anche la condizione di minimalità della somma dei pesi per gli archi scelti, per il grafo dato in input e il relativo parametro k passato come argomento.