

Domanda 2

Non valutata

Contrassegna domanda

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.

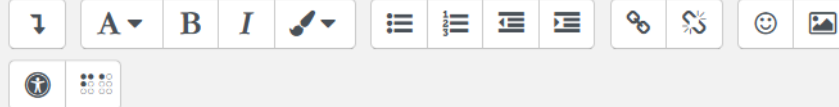
La matrice M' sia allocata dentro alla funzione.

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:

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



Attenzione! Si consiglia l'uso degli spazi al posto delle tabulazioni per l'indentazione del codice, dal momento che il carattere TAB viene utilizzato per la navigazione della pagina da parte della piattaforma.

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

Domanda 3

Non valutata

Contrassegna domanda

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 da almeno k elementi.

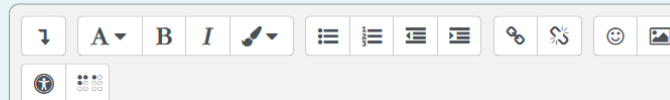
Non è ammesso l'uso di funzioni di libreria.

Esempio:

$a \longleftrightarrow a \longleftrightarrow b \longleftrightarrow b \longleftrightarrow b \longleftrightarrow c \longleftrightarrow d \longleftrightarrow d$

diventa, per $k = 3$

$a \longleftrightarrow a \longleftrightarrow b \longleftrightarrow c \longleftrightarrow d \longleftrightarrow d$



Attenzione! Si consiglia l'uso degli spazi al posto delle tabulazioni per l'indentazione del codice, dal momento che il carattere TAB viene utilizzato per la navigazione della pagina da parte della piattaforma.

```
LIST.h
//LIST.h
//scrivere qui il codice...

LIST.c
//LIST.c
//scrivere qui il codice...
```

Domanda 4

Non valutata

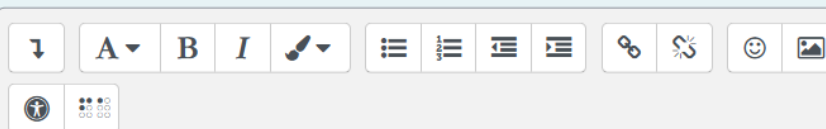
Contrassegna domanda

E' dato un elenco di parole senza duplicati (vettore di stringhe).

Le stringhe in elenco posso essere utilizzate per generare stringhe più lunghe mediante concatenazione. Una concatenazione è una stringa generata dalla sequenza ordinata di tutte o parte delle stringhe in elenco prese al più una volta.

Si scriva una funzione `BESTbestConcat`, che, dato l'elenco, generi la stringa più lunga ottenibile mediante concatenazione, rispettando un vincolo: date due parole consecutive nella sequenza, se la prima termina con vocale, la seconda non può iniziare con vocale. Se la prima termina con consonante, la seconda non può iniziare con consonante. Ai fini della bontà del risultato la lunghezza si misura in numero di caratteri (quindi non di parole).

Nota bene: le due domande a seguire **NON** sono facoltative. Assicurarsi di rispondere a tutte le richieste.



Attenzione! Si consiglia l'uso degli spazi al posto delle tabulazioni per l'indentazione del codice, dal momento che il carattere TAB viene utilizzato per la navigazione della pagina da parte della piattaforma.

```
//scrivere qui il codice...
```

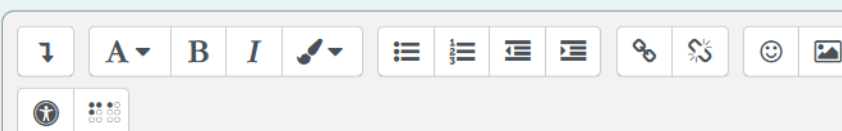
```
char *BESTbestConcat (char **parole, int nparole) {  
}
```

Domanda 5

Non valutata

Contrassegna domanda

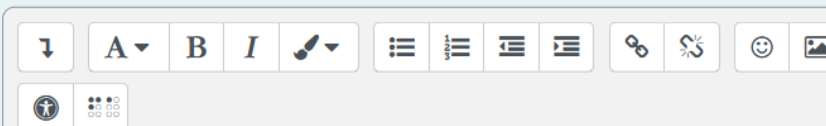
Si giustifichi la scelta del modello combinatorio adottato.

**Domanda 6**

Non valutata

Contrassegna domanda

Si descrivano i criteri di pruning adottati o il motivo della loro assenza.



Informazione

Contrassegna domanda

Descrizione del problema

Dato un grafo non orientato e pesato $G = (V, E)$, si definisce *k-capacitated tree partition* una collezione E_1, E_2, \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 $e \in E$. Sia k un parametro del problema.

Richieste del problema

A seguire una sintesi delle richieste del problema. Per ogni richiesta si troverà una domanda dedicata nelle sezioni a seguire con una descrizione più dettagliata per le richieste.

Strutture dati e letture

Definire opportune strutture dati per rappresentare i dati del problema e tutte le strutture dati ausiliarie ritenute opportune per la risoluzione dei problemi di verifica e di ricerca. Definire inoltre le funzioni di lettura secondo il formato dei file descritti nelle domande a seguire.

Problema di verifica

Data una proposta di partizionamento, valutare che questa rispetti la definizione fornita in precedenza, trascurando l'aspetto di minimalità della somma dei pesi. È sufficiente che la soluzione proposta sia un partizionamento valido.

Problema di ottimizzazione

Identificare, se possibile, un k -capacitated tree partitioning che soddisfi anche la condizione di minimalità della somma dei pesi.

Domanda 7

Non valutata

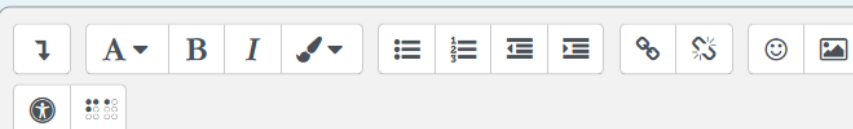
Contrassegna domanda

Strutture dati e acquisizione

Scrivere qui la definizione e implementazione delle strutture dati ritenute 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 riportanti coppie (u, v) di interi, con $0 \leq u, v < V$ a rappresentare gli archi del grafo, e il relativo peso $w_{u,v}$



Attenzione! Si consiglia l'uso degli spazi al posto delle tabulazioni per l'indentazione del codice, dal momento che il carattere TAB viene utilizzato per la navigazione della pagina da parte della piattaforma.

...

Domanda 8

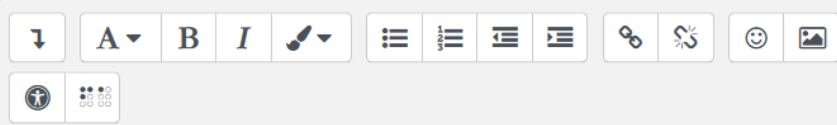
Non valutata

Contrassegna domanda

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.



Attenzione! Si consiglia l'uso degli spazi al posto delle tabulazioni per l'indentazione del codice, dal momento che il carattere TAB viene utilizzato per la navigazione della pagina da parte della piattaforma.

...

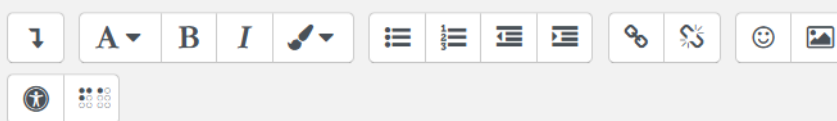
Domanda 9

Non valutata

Contrassegna domanda

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.



Attenzione! Si consiglia l'uso degli spazi al posto delle tabulazioni per l'indentazione del codice, dal momento che il carattere TAB viene utilizzato per la navigazione della pagina da parte della piattaforma.

...

PARTE TEORIA (12 PUNTI)

Domanda 1

Punteggio max.:
2,00

Contrassegna
domanda

Si risolva la seguente equazione alle ricorrenze mediante il metodo dello sviluppo (unfolding):

$$T(n) = T(n-3) + T(n-2) + T(n-1) + 1 \quad n \geq 3$$

$$T(0) = 0 \quad T(1) = 1 \quad T(2) = 2$$

Non trattandosi di una forma standard di equazione alle ricorrenze, si indichi a quale forma la si riconduce e per quali ragioni:

$T(n) =$

Si riporti il primo passo di unfolding:

Si riporti il secondo passo di unfolding:

Si riporti $T(n)$ come sommatoria:

limite inferiore =

limite superiore =

termine della sommatoria =

Si riporti quale è il limite superiore l'ascolto individuato

$T(n) = O($)

Domanda 2

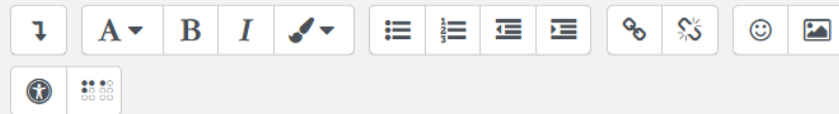
Punteggio max.:
2,00

Contrassegna
domanda

Si determini mediante un algoritmo greedy un codice di Huffman ottimo per i seguenti simboli con le frequenze specificate:

B: 4 C: 10 D: 12 F: 11 J: 14 L: 15 R: 8 X: 3 Z: 6.

Il ramo sinistro abbia etichetta 0, quello destro etichetta 1.



Quali simboli contiene l'aggregato con frequenza 13?

Quali simboli contiene l'aggregato con frequenza 33?

Quale è la codifica di X?

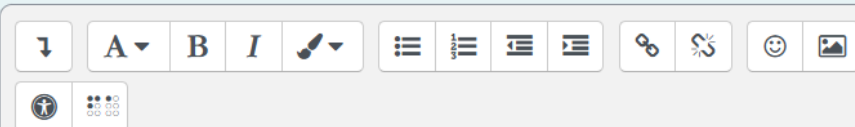
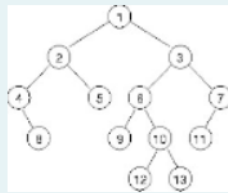
Quale è la decodifica di 011100101110011111011?

Domanda 3

Punteggio max.:
2,00

Contrassegna
domanda

Si visiti in pre-order, in-order e post-order il seguente albero binario:



Formato della risposta:

pre-order:

elenco chiavi

el1, el2, el3, ... eln

in-order:

elenco chiavi

el1, el2, el3, ... eln

post-order:

elenco chiavi

el1, el2, el3, ... eln

Domanda 4

Punteggio max.:
2,00

Contrassegna
domanda

Sia data la sequenza di chiavi intere 13 181 98 110 45 206 167 202 . La si inserisca in una tabella di hash di dimensione 7, inizialmente supposta vuota. Si usi il linear chaining con inserimento in testa a ciascuna lista.

↴

A ▾

B

I

✎ ▾

☰

☰

☰

☰

🔗

🔄

😊

🖼️

👤

👥

Contenuto della lista all'indice 0:

Contenuto della lista all'indice 1:

Contenuto della lista all'indice 2:

Contenuto della lista all'indice 3:

Contenuto della lista all'indice 4:

Contenuto della lista all'indice 5:

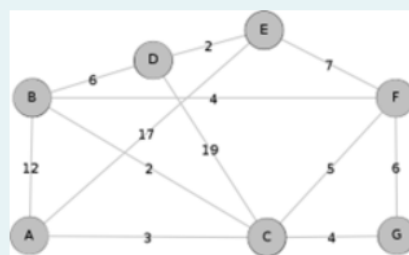
Contenuto della lista all'indice 6:

Domanda 5

Punteggio max.:
2,00

Contrassegna
domanda

Si determini mediante l'algoritmo di Kruskal un albero ricoprente minimo per il grafo non orientato, pesato e connesso in figura.



↴

A ▾

B

I

✎ ▾

☰

☰

☰

☰

🔗

🔄

😊

🖼️

👤

👥

Formato della risposta:

Indicare se gli archi seguenti appartengono all'MST o meno:

- (B,C): sì/no
- (D,E): sì/no
- (B,F): sì/no
- (C,G): sì/no
- (B,D): sì/no
- (F,G): sì/no

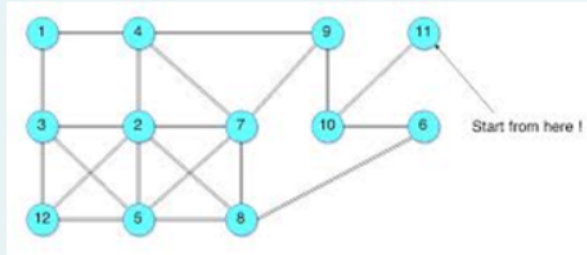
Peso minimo:

Domanda 6

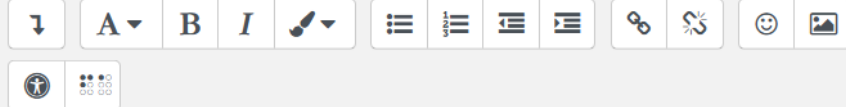
Punteggio max.:
2,00

Contrassegna
domanda

Dato il seguente grafo non orientato



se ne effettui una visita in ampiezza a partire dal vertice **11**. Qualora necessario, i considerino i vertici in ordine numerico crescente.



- quali sono i vertici a distanza 3 dal vertice 11?
- quali sono i vertici a distanza 5 dal vertice 11?
- chi è il padre del vertice 6 nell'albero della visita in ampiezza?
- chi è il padre del vertice 4 nell'albero della visita in ampiezza?