

03MNO Algoritmi e Programmazione

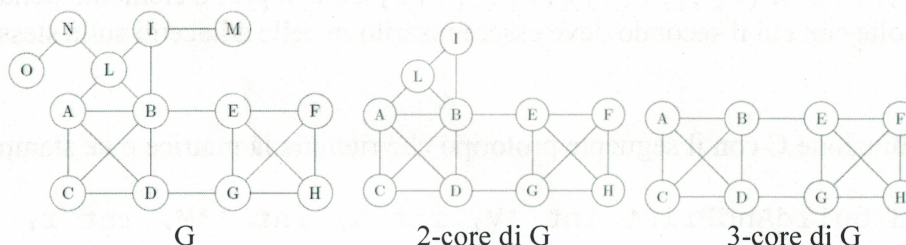
Appello del 28/01/2020 - Prova di programmazione (18 punti)

1. (18 punti)

Nella Teoria dei Grafi, dato un grafo $G = (V, E)$ non orientato, semplice e connesso:

- si definisce **k-core** un sottografo massimale di vertici di G tale che ognuno di questi abbia grado $\geq k$. Vanno considerati unicamente gli archi appartenenti al sottografo.

Esempio:

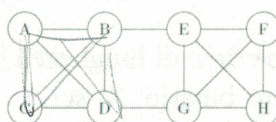


Algoritmo:

per calcolare il k -core di G si cancellino tutti i suoi vertici con grado $< k$. Si noti che la rimozione di un vertice comporta la rimozione di tutti gli archi su di esso incidenti e questo, a sua volta, diminuisce il grado di altri vertici, che quindi possono a loro volta dover essere cancellati. Nell'esempio, se $k=2$, la rimozione del vertice O di grado 1 rende di grado 1 anche il vertice N , originariamente di grado 2, che quindi deve a sua volta essere rimosso. La rimozione del vertice M di grado 1 rende di grado 2 il vertice I , originariamente di grado 3, che quindi non deve a sua volta essere rimosso dal 2-core, ma che sarà rimosso dal 3-core.

- il grafo G si dice **j-edge-connected** se e solo se è necessario rimuovere almeno j archi per sconnetterlo.

Esempio: il seguente grafo è 2-edge-connected: rimuovendo gli archi (B, E) e (D, G) il grafo diventa non connesso. Rimuovendo un solo arco (qualsiasi) il grafo resta connesso.



Un grafo non orientato, semplice e connesso $G = (V, E)$ è memorizzato in un file mediante l'elenco dei suoi archi con il seguente formato: sulla prima riga compare un intero che indica il numero $|V|$ di vertici del grafo, seguono $|V|$ righe ciascuna delle quali riporta l'identificatore alfanumerico di un vertice (max 10 caratteri), segue un numero indefinito di coppie di identificatori che rappresentano gli archi del grafo. Si assuma corretto il formato del grafo su file.

Si scriva un programma C che, ricevuto il nome di un file con la descrizione del grafo G come argomento sulla riga di comando:

1. legga il grafo G e lo memorizzi in un'opportuna struttura dati
2. legga da tastiera un intero $k \geq 0$ e calcoli, se esiste, il **k-core** di G , visualizzando l'elenco dei vertici che vi appartengono
3. legga da tastiera un intero $j \geq 1$ e verifichi se G sia **j-edge-connected**, escludendo l'esistenza di insiemi di archi di cardinalità $< j$ in grado di sconnetterlo e individuando almeno un sottoinsieme di archi di cardinalità j che lo sconnetta.

Osservazioni e suggerimenti:

si suggerisce, qualora sia necessario cancellare vertici, di marcare nel grafo originale i vertici rimossi, piuttosto che eliminarli dalla struttura dati. Di conseguenza si suggerisce di modificare l'ADT di I classe del grafo per tener conto di questa cancellazione "logica".

PER ENTRAMBE LE PROVE DI PROGRAMMAZIONE (18 o 12 punti):

- indicare nell'elaborato e nella relazione nome, cognome e numero di matricola.
- se non indicato diversamente, è consentito utilizzare chiamate a funzioni standard, quali ordinamento per vettori, funzioni su FIFO, LIFO, liste, BST, tabelle di hash, grafi e altre strutture dati, considerate come librerie esterne.
- gli header file devono essere allegati all'elaborato (il loro contenuto riportato nell'elaborato stesso). Le funzioni richiamate, inoltre, dovranno essere incluse nella versione del programma allegata alla relazione. I modelli delle funzioni ricorsive non sono considerati funzioni standard.
- consegna delle relazioni (per entrambe le tipologie di prova di programmazione): entro venerdì 31/01/2020, alle ore 14:00, mediante caricamento su Portale. Le istruzioni per il caricamento sono pubblicate sul Portale nella sezione Materiale). **QUALORA IL CODICE CARICATO CON LA RELAZIONE NON COMPILI CORRETTAMENTE, VERRÀ APPLICATA UNA PENALIZZAZIONE.** Si ricorda che la valutazione del compito viene fatta, senza la presenza del candidato, sulla base dell'elaborato svolto in aula. Non verranno corretti i compiti di cui non sarà stata inviata la relazione nei tempi stabiliti.