



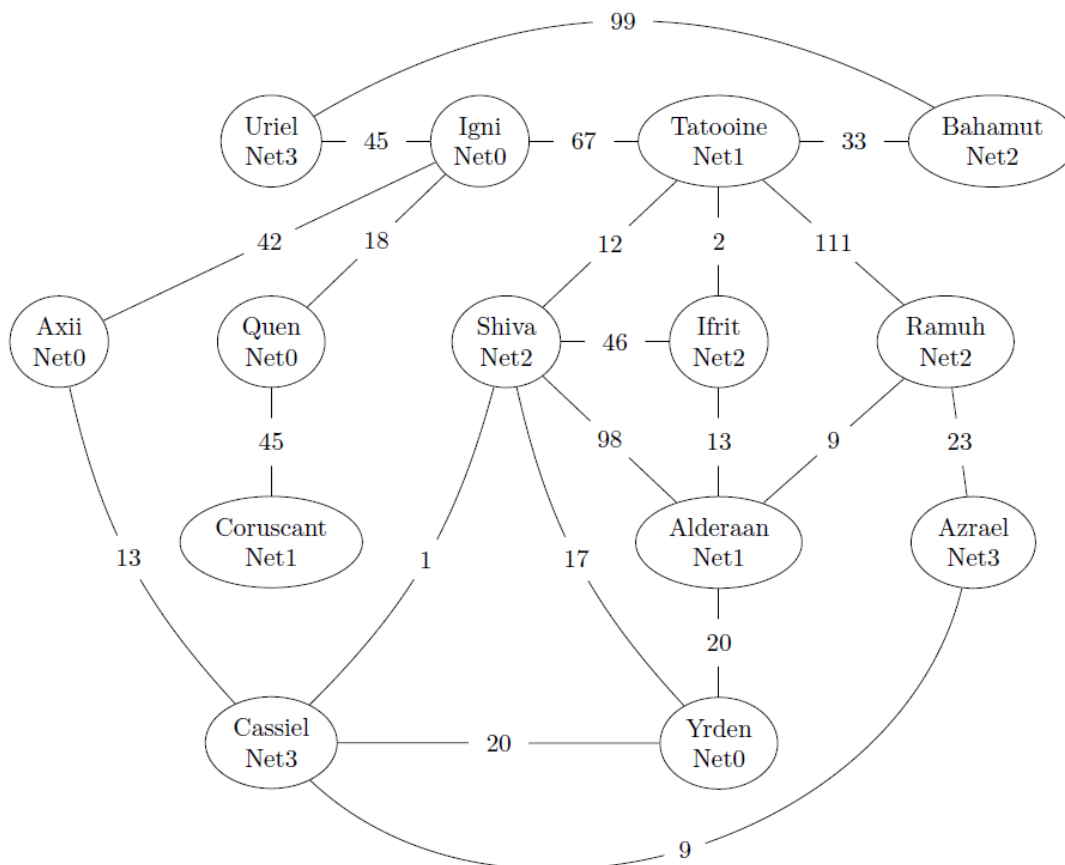
Esercitazione di laboratorio n.12

Laboratorio **NON** valutato

Caricamento nella Sezione Elaborati del Portale entro e non oltre le 23:59 del 12/01/2016

Esercizio n. 1: Grafi - memorizzazione e gestione

Un grafo non orientato e pesato rappresenta una rete di elaboratori appartenenti ciascuno ad una sottorete. Il peso associato ad ogni arco rappresenta il flusso di dati tra due elaboratori della stessa sottorete o di sottoreti diverse, come nell'esempio seguente:



Il grafo è contenuto in un file, il cui nome è passato come argomento sulla linea di comando di un programma C.

Il file ha il seguente formato:

- sulla prima riga un unico intero N rappresenta il numero di vertici del grafo
- seguono N righe ciascuna delle quali contiene una coppia di stringhe di al massimo 30 caratteri, la prima per il nome dell'elaboratore, la seconda per quello della sottorete di appartenenza
- sulle righe successive, in numero indefinito, si trovano coppie di elaboratori ciascuna rappresentata da una coppia di stringhe di lunghezza massima pari a 30 caratteri. Ogni coppia è un arco del grafo. Vi è inoltre, sempre sulla stessa riga, un intero che rappresenta il flusso di dati tra i due elaboratori, cioè il peso dell'arco.



Si facciano anche le seguenti assunzioni:

- i nomi dei singoli nodi sono univoci all'interno del grafo
- non sono ammessi cappi
- tra due nodi c'è al massimo un arco (non è un multigrafo, cfr lucidi Cap. 13 pag. 36).

Si consideri come esempio (parziale) del grafo di figura il seguente file di testo:

```
14
Uriel Net3
Igni Net0
Tatooine Net1
...
Yrden Net0
Igni Uriel 45
Uriel Bahamut 99
Shiva Alderaan 98
...
Yrden Cassiel 20
```

Si scriva un programma in C in grado di caricare in memoria il grafo, leggendone i contenuti da file e di potervi effettuare alcune semplici operazioni.

La rappresentazione della struttura dati in memoria deve essere fatta tenendo conto dei seguenti vincoli:

- il grafo sia implementato come ADT di I categoria, predisposto in modo tale da poter contenere sia la matrice che le liste di adiacenza (nella fase di caricamento dei dati da file si generino le sole liste di adiacenza, su comando esplicito va generata anche la matrice di adiacenza)
- le informazioni relative agli elaboratori e alle sottoreti siano memorizzate in un vettore di Item, dove ciascun item contiene le due stringhe associate (max 30 caratteri).
- il grafo sia realizzato identificando ogni nodo con un indice intero. Gli indici siano generati automaticamente man mano che si acquisiscono i dati. A tale scopo si utilizzi una tabella di simboli (ADT di I categoria) tale da fornire corrispondenze nome->indice e indice->nome. Internamente all'ADT le conversioni nome->indice siano gestite mediante un BST o una tabella di HASH, mentre le corrispondenze indice->nome mediante un vettore ad accesso diretto.

Sul grafo, una volta acquisito da file, sia possibile:

- calcolare e visualizzare il nome di ogni nodo
- stampare il numero totale di vertici ed elencarli esplicitamente per nome
- stampare il numero di archi incidenti su un nodo e l'elenco di vertici ad esso connessi
- generare la matrice di adiacenza, **SENZA** leggere nuovamente il file, a partire dalle liste di adiacenza
- determinare l'ammontare complessivo dei flussi intra-rete ed inter-rete globali. Si noti che per ottenere questi due valori è sufficiente percorrere iterativamente tutti gli archi del grafo.

Esercizio n. 2: Società sportiva (*Tema d'esame del 21/9/2012*)

Una società sportiva deve organizzare un programma di allenamento per il potenziamento delle prestazioni fisiche dei propri atleti. Il programma prevede quattro fasi di allenamento. Per ogni fase di allenamento è disponibile un certo numero di esercizi, ognuno dei quali caratterizzato da un determinato dispendio energetico. Il numero massimo di esercizi impiegabili in ciascuna fase è pari a n ed il dispendio massimo è di m calorie (valori interi inseriti dall'utente). Si noti che un esercizio



può comparire in più fasi, cioè potrebbe essere potenzialmente selezionato per più fasi. Nella soluzione invece ogni esercizio compare in al massimo una fase.

Le informazioni relative agli esercizi disponibili, al dispendio energetico loro associato (espresso tramite un numero intero minore o uguale a 500) e alle fasi in cui sono impiegabili sono memorizzate in un file di testo. Per ogni esercizio, nel file è riportata una riga contenente il nome dell'esercizio (stringa senza spazi di al più 20 caratteri), il dispendio energetico associato, il numero di fasi in cui può essere impiegato e quindi l'elenco di tali fasi (ogni fase è identificata da un numero intero compreso tra 1 e 4). Il numero totale di esercizi disponibili è riportato sulla prima riga del file.

Si realizzi un programma in linguaggio C che, una volta letti da tastiera il valore di n e m , e a partire dal contenuto del file suddetto, sia in grado di determinare per ciascuna fase al più n esercizi da svolgere in modo tale da avvicinarsi il più possibile al dispendio energetico massimo fissato m , senza però superarlo. Si tenga presente che lo stesso esercizio non può essere svolto in più di una fase. Il programma deve stampare a video, per ognuna delle quattro fasi, la lista degli esercizi ed il dispendio energetico loro associato.

Si implementi anche una seconda funzionalità che, letta da file una proposta di soluzione (proposta in un formato a scelta), determini se questa è valida (ossia, se rispetta tutti i vincoli sopra elencati).

Si assuma che il contenuto del file di ingresso sia quello riportato di seguito (a sinistra):

13						FASE 1	
femorali	380	3	2	3	4	glutei	260
deltoidi	230	2	1	3		adduttori	340
tricipiti	140	3	1	3	4	bicipiti	200
abduuttori	190	2	2	3		FASE 2	
pettorali	320	2	2	4		gastrocnemio	220
dorsali	350	2	3	4		abduuttori	190
trapezi	300	1	2			femorali	380
quadricipiti	150	2	2	4		FASE 3	
bicipiti	200	2	1	2		dorsali	350
adduttori	340	3	1	3	4	tricipiti	140
glutei	260	2	1	3		deltoidi	230
addominali	470	3	1	2	4	FASE 4	
gastrocnemio	220	2	1	2		addominali	470
						pettorali	320

Sulla destra, dunque, è riportata una possibile soluzione nel caso in cui l'utente inserisse i valori $n=3$ e $m=800$.