Laboratorio di Algoritmi e Strutture Dati

Docenti: M. Torelli, S. Aguzzoli

Appelli del 29 gennaio e del 19 febbraio 2009
Progetto “FACCIALIBRO”

Introduzione

Il progetto è ispirato ad un popolarissimo sito di *social networking* (in italiano, rete sociale). Per *rete sociale* si intende una comunità online di persone che condividono interessi o attività e che sono motivate a conoscere e gli interessi e le attività delle altre persone.

Obiettivo del progetto è sviluppare un sistema che supporti lo sviluppo di una rete sociale: chiameremo tale sistema FACCIALIBRO.

Organizzazione degli appelli

Il presente progetto è valido per gli appelli del 29 gennaio 2009 e del 19 febbraio 2009.

- Per l’appello del 29 gennaio la data ultima di consegna è fissata per il 18 febbraio 2009 incluso.
- Per l’appello del 19 febbraio la data ultima di consegna è fissata per il 9 marzo 2009 incluso.

Chi consegna dopo il 18 febbraio è tenuto a implementare anche le specifiche riportate nella sezione **Funzionalità aggiuntive**. Tali specifiche sono solo facoltative per coloro che consegnano entro il 18 febbraio.

La realizzazione del progetto è una prova d’esame da svolgersi **individualmente**. I progetti giudicati frutto di **collaborazioni** saranno **estromessi** d’ufficio dalla valutazione.

Il problema

Le caratteristiche fondamentali di FACCIALIBRO sono le seguenti: ogni persona è rappresentata da un *profilo*; due profili possono essere legati da una *relazione di amicizia*; ogni profilo può aderire a *gruppi* che condividono interessi o attività. Il sistema inoltre fornisce alcune funzionalità particolari, ad esempio consiglia nuove amicizie e calcola il grado di separazione tra due profili.

Ogni profilo interagisce con il sistema attraverso *comandi di profilo* prestabiliti. Per consentire la gestione del sistema, esistono poi altri *comandi di sistema* speciali.

Profili

Ogni partecipante di FACCIALIBRO è rappresentato da un *profilo*, caratterizzato da due elementi:

- il *nickname*: stringa univoca di lunghezza arbitraria, priva di spazi;
- lo *status*: stringa di lunghezza arbitraria, di solito contenente una frase in terza persona che denota lo stato d’animo o l’azione che il profilo sta svolgendo.

Il nickname va scelto al momento della creazione del profilo, lo status può essere modificato in ogni momento attraverso un comando di profilo.

Il sistema deve essere in grado di elencare tutti i profili di FACCIALIBRO, e stampare lo status di un profilo individuato da un dato nickname.

Nel seguito, con un abuso di linguaggio, identificheremo ogni profilo (e quindi ogni partecipante a FACCIALIBRO da esso rappresentata) con il proprio nickname.

Gruppi

Ogni partecipante di FACCIALIBRO può creare un *gruppo* che raccoglie partecipanti con interessi comuni. Un gruppo è individuato da un titolo, che è una stringa univoca di lunghezza arbitraria. Ogni partecipante di FACCIALIBRO può aderire ad un *gruppo* già esistente (e quindi diventarne *membro*) e, in un secondo momento, disiscriversi dal gruppo.

Il sistema deve essere in grado di sapere quali gruppi sono attivi, a quali gruppi aderisce ciascun partecipante, quali membri ha ciascun gruppo.

Amici

Due partecipanti di FACCIALIBRO possono dichiarare al sistema di essere amici. La relazione di amicizia è simmetrica e deve essere dichiarata da entrambi gli amici, anche in momenti diversi.

Il sistema deve essere in grado di dire se due partecipanti sono legati da una relazione di amicizia e ricordare le dichiarazioni di amicizia non ancora ricambiate.

Se p e q sono due amici, allora p può chiedere al sistema lo status di q, l'elenco dei suoi amici e i gruppi di cui q è membro.

Grado di Separazione

Siano p e q due partecipanti di FACCIALIBRO. Il *grado di separazione* tra p e q è il minimo intero n tale che esistono profili $r_0, r_1, \ldots, r_n$ con $r_0 = p$, $r_n = q$ e r_i amico di r_{i+1} per ogni $i = 0, 1, \ldots, n-1$. Se tale intero n non esiste allora il grado di separazione tra p e q è infinito.

Sia G un gruppo. Il *grado di separazione relativo a G* tra p e q è il minimo intero n tale che esistono profili $r_0, r_1, \ldots, r_n$ con r_j membro di G per ogni $j = 0, 1, \ldots, n$; $r_0 = p$, $r_n = q$; r_i amico di r_{i+1} per ogni $i = 0, 1, \ldots, n-1$. Se tale intero n non esiste allora il grado di separazione relativo a G tra p e q è infinito.

Il sistema deve essere in grado di calcolare i gradi di separazione tra due profili.

Segnalazione di nuovi amici

Un partecipante p di FACCIALIBRO può chiedere al sistema di segnalargli profili che potrebbe riconoscere come amici. La regola usata dal sistema è la seguente: un partecipante q diverso da p e non ancora amico di p viene segnalato a p se esistono almeno tre partecipanti distinti da p e q che sono amici sia di p che di q.

Esempio

Immaginiamo che FACCIALIBRO sia costituito dai seguenti profili:

- nickname: `anna.bella` - status: `sta preparando una torta`
- nickname: `djmirkoooo` - status: `sta ascoltando il nuovo cd dei Uoki Tochi`
- nickname: `franco83` - status: `e' solo e cerca amici`
- nickname: `marco70` - status: `ha bisogno di parlare con qualcuno`
- nickname: `martamatta` - status: `non vede l'ora di andare alla scuola di circo`

legati dalle seguenti relazioni di amicizia:

- `anna.bella` è amica di `djmirkoooo` e `marco70`
- `djmirkoooo` è amico di `anna.bella`

- `franco83` ha dichiarato di essere amico di `anna.bella`, ma lei non ha fatto altrettanto (quindi non sono ancora amici)
- `marco70` è amico di `anna.bella`

e partecipanti ai seguenti gruppi:

- `Ricette` di cui fanno parte `anna.bella` e `martamatta`
- `Persone che usano il cellulare per vedere al buio` di cui fanno parte `franco83`, `djmirkoooo`, `marco70` e `martamatta`

Il grado di separazione tra `marco70` e `djmirkoooo` è 2. Il grado di separazione tra `marco70` e `martamatta` è infinito. Il sistema al momento non può segnalare alcuna nuova amicizia. Se ora supponiamo che sia `franco83` sia `martamatta` siano diventati amici di `anna.bella`, `djmirkoooo` e `marco70`, allora il sistema può segnalare `martamatta` a `franco83`.

Specifiche di implementazione

Il programma deve leggere dallo standard input (`stdin`) una sequenza di righe (separate da `\n`), ciascuna delle quali corrisponde a un comando. Quando un comando è letto, deve essere eseguito immediatamente. Le operazioni di stampa sono effettuate sullo standard output (`stdout`), e ogni stampa deve iniziare su una nuova riga.

Sono previsti due tipi di comandi, entrambi testuali:

- comandi di profilo: iniziano con un nickname e si intendono impartiti al sistema dal partecipante con quel nickname;
- comandi di sistema: iniziano con la parola riservata `system`.

La sintassi e la semantica dei comandi è specificata nelle Tabelle 1,2 e 3, dove `nick`, `nick2`, `st` e `group` sono stringhe di lunghezza arbitraria[1] sull'alfabeto dei caratteri ASCII stampabili (*i.e.* l'insieme di tutti i caratteri `c` tali che `isprint(c)` è maggiore di 0). In particolare `nick` e `nick2` sono prive di spazi.

Si noti che i comandi possono essere implementati liberamente; in particolare, non vi è necessaria corrispondenza tra comandi e prototipi di funzioni.

Tabella 1: Comandi di sistema

Comando	**Descrizione**
`system p`	Il sistema stampa l'elenco di tutti i profili.
`system g`	Il sistema stampa l'elenco di tutti i gruppi.
`system ! nick nick2`	Il sistema stampa il grado di separazione tra `nick` e `nick2`. Se non esiste alcun profilo con nickname `nick` o `nick2`, allora il sistema non fa niente.
`system ? nick nick2 group`	Il sistema stampa il grado di separazione relativo a `group` tra `nick` e `nick2`. Se non esiste alcun profilo con nickname `nick` o `nick2`, oppure non esiste il gruppo con titolo `group`, allora il sistema non fa niente.
`system q`	Il sistema termina l'esecuzione.

[1] ossia, non deve essere definita nel programma alcuna costante k che limiti la lunghezza di una stringa a k.

Tabella 2: Comandi di profilo

Comando	Descrizione
`nick n st`	Con questo comando `nick` modifica il proprio status. Se non esiste alcun profilo con nickname `nick`, allora il sistema crea un nuovo profilo con nickname `nick` e status `st`. In caso contrario, rimpiazza con `st` il vecchio status di `nick`.
`nick d`	Con questo comando `nick` cancella il proprio profilo. Vengono rimosse tutte le informazioni riguardanti `nick`. In particolare vengono cancellate anche le richieste di amicizia ricevute da `nick` e non ancora ricambiate. Se non esiste alcun profilo con nickname `nick`, il sistema non fa niente.
`nick a nick2`	Con questo comando `nick` dichiara al sistema che `nick2` è un suo amico. Se non esiste un profilo con nickname `nick` oppure non esiste un profilo con nickname `nick2`, il sistema non fa niente. In caso contrario, se in precedenza `nick2` ha già dichiarato che `nick` è suo amico, allora il sistema lega i due profili nella relazione di amicizia; in caso contrario il sistema memorizza soltanto questa nuova dichiarazione.
`nick c`	Con questo comando `nick` chiede al sistema di consigliargli nuovi amici. Se non esiste un profilo con nickname `nick` il sistema non fa niente. In caso contrario, il sistema stampa i nickname di tutti i possibili nuovi amici di `nick` secondo la regola specificata sopra.
`nick S nick2`	Con questo comando `nick` chiede al sistema lo status di `nick2`. Se non esiste un profilo con nickname `nick` oppure non esiste un profilo con nickname `nick2`, il sistema non fa niente. In caso contrario: se `nick` e `nick2` coincidono o sono amici, allora il sistema stampa lo status di `nick2`; altrimenti il sistema stampa il messaggio di errore: `err - nick:  non autorizzato`
`nick A nick2`	Con questo comando `nick` chiede al sistema quali sono gli amici di `nick2`. Se non esiste un profilo con nickname `nick` oppure non esiste un profilo con nickname `nick2`, il sistema non fa niente. In caso contrario: se `nick` e `nick2` coincidono o sono amici, allora il sistema stampa l'elenco dei nickname degli amici di `nick2`; altrimenti il sistema stampa il messaggio di errore: `err - nick:  non autorizzato`
`nick + group`	Con questo comando `nick` aderisce al gruppo `group`. Se non esiste un profilo con nickname `nick`, il sistema non fa niente. In caso contrario, Se esiste il gruppo con titolo `group`, allora il sistema ne rende membro `nick`; altrimenti il sistema crea un nuovo gruppo con titolo `group` e ne rende membro `nick`.

Funzionalità aggiuntive

Modifichiamo la definizione di *status* come segue. Uno *status* è una stringa della forma `aaaa/mm/gg xxx` dove `aaaa/mm/gg` rappresenta una data e `xxx` è una stringa di lunghezza arbitraria sull'alfabeto dei caratteri ASCII stampabili. La data indica il momento in cui lo status è stato impostato.

Oltre ad implementare tutti i comandi delle Tabelle 1,2 e 3 (attenzione alla nuova definizione di status), si richiede di implementare il comando nella Tabella 4.

Si assume che se la data dello status `st1` precede la data dello status `st2` allora l'immissione del comando `nick n st1` deve precedere l'immissione del comando `nick n st2` (si noti che entrambi i comandi aggiornano lo status dello stesso profilo `nick`). Non si può invece fare alcuna assunzione sull'ordine di immissione dei comandi `nick n st1` e `nick2 n st2` se `nick` è diverso da `nick2`.

Tabella 3: Comandi di profilo - continua

`nick - group`	Con questo comando `nick` si cancella dal gruppo `group`. Se non esiste un profilo con nickname `nick`, il sistema non fa niente. Se non esiste il gruppo con titolo `group`, oppure `nick` non ne fa parte, allora il sistema stampa il messaggio di errore: `err - nick: non fai parte del gruppo group.`

`nick m group`	Con questo comando `nick` chiede di stampare i membri del gruppo `group`.
	Se non esiste un profilo con nickname `nick`, il sistema non fa niente. In caso contrario: se `group` è il titolo di un gruppo esistente e `nick` ne fa parte, allora il sistema stampa i nickname di tutti i membri del gruppo. Se non esistono gruppi con titolo `group`, allora il sistema stampa il messaggio di errore:
	`err - nick: il gruppo group non esiste.`
	Se `nick` non fa parte del gruppo, allora il sistema stampa il messaggio di errore:
	`err - nick: non autorizzato.`

`nick G nick2`	Con questo comando `nick` chiede al sistema a quali gruppi aderisce `nick2`. Se non esiste un profilo con nickname `nick` oppure non esiste un profilo con nickname `nick2`, il sistema non fa niente. In caso contrario: se `nick` e `nick2` coincidono o sono amici, allora il sistema stampa l'elenco dei gruppi cui aderisce `nick2`; altrimenti il sistema stampa il messaggio di errore: `err - nick: non autorizzato`

Tabella 4: Funzionalità aggiuntiva

Comando	**Descrizione**
`nick s`	Con questo comando `nick` chiede al sistema di individuare tra i suoi amici il nickname `nick2` di colui che ha impostato il proprio status `st` più recentemente (se più profili soddisfano questa condizione, il sistema ne sceglie uno arbitrariamente). Se non esiste alcun profilo con nickname `nick` oppure `nick` non ha amici, allora il sistema non fa niente. Altrimenti stampa un messaggio della forma: `nick2: st`

Note

1. Non devono essere presenti vincoli sul numero di profili e di gruppi.

2. Per semplicità si suppone che l'input sia sempre conforme alle specifiche delle Tabelle 1,2,3 e 4, per cui non è necessario controllare la correttezza dell'input. Per leggere l'input si usino le funzioni standard ANSI C `getchar()` e/o `scanf()`.

3. **Formato per la stampa di elenchi di nickname**
   Siano $\texttt{nick}_1, \texttt{nick}_2, \ldots, \texttt{nick}_n$ i nickname da stampare, elencati in un ordine arbitrartio. Allora bisogna stampare la linea:
   $(\texttt{nick}_1\ \texttt{nick}_2\ \ldots\ \texttt{nick}_n)$

4. **Formato per la stampa di titoli di gruppi**
   Siano $\texttt{group}_1, \texttt{group}_2, \ldots, \texttt{group}_n$ i titoli dei gruppi da stampare, elencati in un ordine arbitrartio. Allora bisogna stampare quanto segue:
   [
   $\texttt{group}_1$
   $\texttt{group}_2$

```
⋮
groupn
]
```

Esempio

Si supponga che le righe di input siano:

```
Anna n Ciao a tutti!
Bruno n Eccomi qua
Carlo n niente da dire
Davide n sono contento
Davide a Anna
Davide S Anna
Anna + Evviva l'assioma della scelta
system p
system g
Bruno + Evviva l'assioma della scelta
Bruno + tiritere senza senso
Davide a Bruno
Bruno a Carlo
Anna a Carlo
Carlo a Bruno
Bruno S Carlo
Carlo G Bruno
Carlo G Anna
Davide A Anna
Anna a Davide
Davide A Anna
Davide S Anna
Bruno A Davide
Bruno a Davide
Bruno c
Anna n mi annoio
Davide S Anna
elena n sto mangiando il gelato
Anna a elena
elena n adoro le fragole
elena + Evviva l'assioma della scelta
elena a Anna
Bruno a elena
elena a Anna
Bruno - Evviva l'assioma della scelta
elena + dolci e gelati
elena a Anna
elena a Bruno
Carlo d
System p
System g
Carlo n Rieccomi!!!
Carlo a elena
```

```
elena a Carlo
Bruno a Carlo
Anna a Carlo
Carlo a Bruno
Carlo a Anna
Bruno c
Bruno a Anna
Anna a Bruno
elena c
fabio n sono nuovo di qui
system ! fabio Bruno
giorgio n Voglio conoscere qc1
fabio a giorgio
giorgio a fabio
giorgio + Evviva l’assioma della scelta
giorgio a elena
system ! fabio Bruno
elena a giorgio
system ! fabio Bruno
giorgio + tiritere senza senso
giorgio + dolci e gelati
giorgio + Paralipomeni della Batrachiomiomachia
fabio + Paralipomeni della Batrachiomiomachia
elena + Paralipomeni della Batrachiomiomachia
Anna + Paralipomeni della Batrachiomiomachia
Bruno + Paralipomeni della Batrachiomiomachia
system ? Bruno giorgio Paralipomeni della Batrachiomiomachia
Carlo + Paralipomeni della Batrachiomiomachia
system ? Bruno giorgio Paralipomeni della Batrachiomiomachia
fabio G giorgio
Bruno m Paralipomeni della Batrachiomiomachia
giorgio m Evviva l’assioma della scelta
elena S Bruno
elena A Bruno
Anna d
Carlo d
system ! Bruno giorgio
system ? Bruno giorgio tiritere senza senso
system p
system g
system q
```

L’output prodotto dal programma deve essere il seguente

```
err - Davide: non autorizzato
(Anna Bruno Carlo Davide)
[
Evviva l’assioma della scelta
]
niente da dire
[
```

```
tiritere senza senso
Evviva l’assioma della scelta
]
err - Carlo: non autorizzato
err - Davide: non autorizzato
(Davide)
Ciao a tutti!
err - Bruno: non autorizzato
()
mi annoio
(Anna Bruno Davide Elena)
[
dolci e gelati
tiritere senza senso
Evviva l’assioma della scelta
]
(Anna)
()
infinito
infinito
3
2
2
[
Paralipomeni della Batrachiomiomachia
dolci e gelati
tiritere senza senso
Evviva l’assioma della scelta
]
(giorgio fabio elena Anna Bruno Carlo)
(Anna elena giorgio)
Eccomi qua
(Anna Carlo Davide elena)
2
infinito
(Bruno Davide elena fabio giorgio)
[
Paralipomeni della Batrachiomiomachia
dolci e gelati
tiritere senza senso
Evviva l’assioma della scelta
]
```

Funzionalità aggiuntive: Esempio

Si supponga che le righe di input siano:

```
Anna n 2009/01/13 Ciao a tutti!
Bruno n 2008/04/29 Eccomi qua
Carlo n 2009/01/29 niente da dire
Davide n 1492/10/12 sono contento
Anna a Bruno
Bruno a Anna
```

```
Anna a Carlo
Bruno a Carlo
Carlo a Anna
Carlo a Bruno
Davide a Anna
Davide a Carlo
Davide a Bruno
Anna a Davide
Carlo a Davide
Bruno a Davide
Anna s
Carlo s
Davide n 2009/03/03 ci sono ancora!
Anna s
system q
```

L'output prodotto dal programma deve essere il seguente

```
Carlo: 2009/01/29 niente da dire
Anna: 2009/01/13 Ciao a tutti!
Davide: 2009/03/03 ci sono ancora!
```

Presentazione del progetto

Il progetto deve essere inviato per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it`, o entro il 18 febbraio 2009 (incluso), se si vuole sostenere l'orale nell'appello del 29 gennaio, o entro il 9 marzo 2009 (incluso), se si vuole sostenere l'orale nell'appello del 19 febbraio. La discussione del progetto e l'esame orale si svolgeranno in data e luogo da specificarsi (consultare al riguardo il sito: `http://homes.dsi.unimi.it/∼aguzzoli/algo.htm`).

Occorre presentare:

1. il codice sorgente (rigorosamente ANSI C, compilabile con **gcc**);
2. una sintetica relazione (formato pdf o rtf) che illustra le strutture dati utilizzate e analizza il costo delle diverse operazioni richieste dalla specifica.

I due o più file (file sorgenti C + relazione) devono essere contenuti in un unico file `.zip` il cui nome dovrà essere `cognome.zip`. La relazione e il codice devono riportare il vostro nome, cognome e matricola.

Una copia cartacea della relazione e del codice deve inoltre essere consegnata al dr. Aguzzoli entro le scadenze fissate (lasciandola eventualmente nella sua casella postale presso il dipartimento in via Comelico).

Si ricorda infine di presentarsi alla prova orale con una copia stampata della relazione e del codice.

Il calendario degli esami orali sarà disponibile sulla pagina del corso qualche giorno dopo il termine di consegna del progetto.

Per ogni ulteriore chiarimento:
E-mail: `aguzzoli@dsi.unimi.it`
Ricevimento: il mercoledì, ore 15-16, stanza S204.

Avvisi

La versione aggiornata del progetto è pubblicata in .pdf sul sito:
`http://homes.dsi.unimi.it/∼aguzzoli/algo.htm`.
Si consiglia di consultare periodicamente questo sito per eventuali correzioni e/o precisazioni relative al testo del progetto.

Si richiede allo studente di effettuare un adeguato collaudo del proprio progetto su numerosi esempi diversi per verificarne la correttezza e valutarne le prestazioni.

La realizzazione del progetto è una prova d'esame da svolgersi **individualmente**. I progetti giudicati frutto di **collaborazioni** saranno **estromessi** d'ufficio dalla valutazione.