

# Laboratorio di Algoritmi

Docenti: M. Goldwurm, M. Frasca

## Progetto “Facebook”

valido per gli appelli di gennaio e febbraio 2016

### 1. Descrizione

Il progetto richiede la stesura di un programma scritto in linguaggio C che consenta all'utente di gestire una versione semplificata dei contatti *Facebook*. Gli utenti di Facebook si assume siano individuati mediante *nome*, *cognome* e un *identificativo* (intero univoco). Le informazioni degli utenti iniziali devono essere lette nel modo seguente:

- Il programma deve chiedere all'utente qual è il numero  $n$  di utenti iniziali
- Quindi deve leggere dallo standard input le informazioni degli  $n$  utenti, uno per volta, nel seguente formato:

nome\_i cognome\_i identificativo\_i

- dove *nome\_i*, *cognome\_i* e *identificativo\_i* sono rispettivamente nome, cognome ed identificativo dell' $i$ -mo utente inserito,  $1 \leq i \leq n$ . I tre campi sono separati da spazio, mentre ogni riga è terminata dal carattere a capo “\n”.

Gli utenti di Facebook possono instaurare *relazioni di amicizia* tra loro, e ogni utente può avere più relazioni di amicizia. Ogni relazione di amicizia si instaura tra una singola coppia di utenti  $u$  e  $v$ , e in tal caso  $u$  si dirà amico di  $v$  e  $v$  amico di  $u$ . Ad ogni relazione inoltre è associato l'anno (intero) di creazione della stessa.

Le operazioni da implementare, descritte nella Sezione 2, variano a seconda dell'appello per il quale avviene la consegna: per l'appello di gennaio sono obbligatorie le operazioni 2.1, 2.2, 2.3, 2.4, 2.5, mentre sono facoltative le operazioni 2.6 e 2.7; per l'appello di febbraio sono obbligatorie le operazioni 2.1, 2.2, 2.3, 2.6, 2.7, facoltative le operazioni 2.4 e 2.5.

## 2. Operazioni

Il programma deve visualizzare a video le seguenti possibilità di scelta, ritornando a tale visualizzazione dopo ogni operazione effettuata dall'utente:

### 2.1. Inserire un nuovo utente Facebook

Questo comando deve chiedere di inserire da riga di comando il nome, il cognome ed e l'identificativo del nuovo utente Facebook, ed inserire il nuovo utente solo nel caso il suo identificativo non esista già. In caso contrario, il comando deve visualizzare sullo standard output il messaggio “Inserimento fallito, l'identificativo è già presente”.

### 2.2. Cercare un utente Facebook

Questo comando deve chiedere l'identificativo dell'utente Facebook che si vuole cercare e stampare a video le informazioni associate (nome e cognome), nel caso tale identificativo esista, nel seguente formato

nome cognome identificativo

Se l'identificativo non fosse presente, il comando deve visualizzare il messaggio “Utente non trovato”.

### 2.3. Inserire una nuova relazione di amicizia

Il comando deve chiedere gli identificativi  $X$  e  $Y$  dei due utenti Facebook di cui si vuole inserire una nuova relazione di amicizia e l'anno di creazione della relazione stessa. Quindi inserire la nuova relazione nel caso i due utenti esistano. In caso contrario, il comando deve visualizzare il messaggio “Utente  $U$  non presente”, dove  $U$  è l'identificativo dell'utente non trovato (quindi  $X$  o  $Y$ ), oppure, se entrambi gli utenti non esistono, “Utenti  $X$  ,  $Y$  non presenti”.

### 2.4. Determinare i gruppi di amici

Il comando deve determinare, analizzando le relazioni di amicizia esistenti, quei sottoinsiemi di utenti che risultino in relazione di amicizia, diretta o indiretta. Specificamente, l'insieme  $U = \{u_1, u_2 \dots, u_m\}$  di utenti Facebook è un *gruppo di amici* se, per ogni coppia  $1 \leq i, j \leq m$ , si verifica che:

- (amicizia diretta)  $u_i$  è amico di  $u_j$ , oppure

- (amicizia indiretta) esistono  $u_{i_1}, u_{i_2}, \dots, u_{i_k} \in U$ , con  $k \leq m$ ,  $i_1, i_2, \dots, i_k \in \{1, 2, \dots, m\}$ , tali che  $u_i = u_{i_1}$ ,  $u_j = u_{i_k}$  e per ogni  $r \in \{1, 2, \dots, k-1\}$   $u_{i_r}$  è amico di  $u_{i_{r+1}}$ .

I gruppi di amici devono essere visualizzati nel modo seguente:

```

Gruppo i
nome_i1      cognome_i1  identificativo_i1
nome_i2      cognome_i2  identificativo_i2
.....

```

## 2.5. Determinare i gruppi di amici nati dall'anno x

Come per il punto 2.4, il comando deve determinare i gruppi di amici Facebook, ma considerando solo le relazioni di amicizia nate dall'anno x in poi, escludendo le altre. L'anno x deve essere inserito dalla linea di comando.

## 2.6. Determinare la 'debolezza' di ogni relazione di amicizia

Data una relazione di amicizia  $(u, v)$ , per *debolezza*  $d(u,v)$  definiamo la seguente quantità:

$$d(u, v) = 1 - \frac{|N_u \cap N_v|}{|N_u \cup N_v|}$$

dove  $N_x$  è l'insieme degli amici (diretti) dell'utente x. Più elevato è il numero di amici che i due utenti hanno in comune, minore sarà la debolezza della loro relazione. Il comando deve calcolare la debolezza di ogni relazione di amicizia esistente, e visualizzare in colonna per ogni relazione  $(u, v)$  il seguente output:

```
u - v ----- d(u, v)
```

## 2.7. Determinare il coefficiente di amicizia tra due utenti

Il comando assume che sia già stato eseguito il comando 2.6 e si conoscano già gli indici di debolezza di ogni relazione. Il comando deve chiedere di inserire da linea di comando gli identificativi dei due utenti  $u$  e  $v$  di cui si vuole calcolare il coefficiente di amicizia e anzitutto controllare che  $u \neq v$ . Il coefficiente di amicizia è definito come segue.

- (catena di amici) Una *catena di amici* tra  $u$  e  $v$  è una sequenza ordinata di utenti  $c_{uv} = (w_1, w_2, \dots, w_k)$ , dove  $k$  è un intero positivo al più pari al numero di

utenti esistenti,  $u = w_1$ ,  $v = w_k$  e per ogni  $r \in \{1, 2, \dots, k-1\}$   $w_r$  è amico di  $w_{r+1}$ .

- (Debolezza di una catena) Data una catena  $c = (w_1, w_2, \dots, w_k)$ , si definisce *debolezza della catena*  $T(c)$  la somma degli indici di debolezza di tutte le

relazioni di amicizia della catena, cioè  $T(c) = \sum_{i=1}^{k-1} d(w_i, w_{i+1})$

- (coefficiente di amicizia) Il *coefficiente di amicizia*  $A_{uv}$  tra gli utenti  $u$  e  $v$  è un indice di quanto deboli siano le catene tra due utenti  $u$  e  $v$ , ed è definito come segue:

$$A_{uv} = \frac{1}{\min_{c \in C_{uv}} T(c) + 1}$$

dove  $C_{uv}$  è l'insieme di tutte le catene di amicizia esistenti tra  $u$  e  $v$ . Una volta calcolato  $A_{uv}$ , il comando deve stamparlo a video, e nelle righe seguenti deve stampare tutte le relazioni della catena corrispondente, una relazione di amicizia per ogni riga, con il relativo indice di debolezza. Un esempio di output del comando è il seguente:

Il coefficiente di amicizia tra  $u$  e  $v$  è  $A_{uv}$

```

w1 - w2  -----  d(w1, w2)
w2 - w3  -----  d(w2, w3)
... ..
wk-1 - wk -----  d(wk-1, wk)

```

Terminata una operazione il programma deve rimanere in attesa che l'utente chieda una nuova operazione, oppure chieda di uscire digitando “esci”. Le operazioni vanno implementate utilizzando le strutture dati e gli algoritmi più appropriati ed efficienti.

Il progetto deve essere svolto individualmente. I progetti giudicati frutto di copiatura saranno opportunamente sanzionati. Si richiede allo studente di effettuare un adeguato collaudo del proprio progetto su numerosi esempi per verificarne la correttezza.

La versione aggiornata del progetto è pubblicata in formato pdf sul sito:

**<http://frasca.di.unimi.it/DILASD16.html>**

Si consiglia di consultare periodicamente questo sito per eventuali correzioni e/o precisazioni relative al testo del progetto. Per ogni ulteriore chiarimento scrivere una e-mail all'indirizzo [frasca@di.unimi.it](mailto:frasca@di.unimi.it).

Il presente progetto è valido per gli appelli del 29 gennaio e del 2 marzo 2016, e deve essere consegnato **entro le 12:00 del 21 gennaio 2016** per l'appello di gennaio, ed **entro le 12:00 del 20 febbraio 2016** per l'appello di febbraio.

Le valutazioni saranno disponibili sulla pagina del corso nei giorni successivi al termine di consegna del progetto. Il progetto va inviato per posta elettronica all'indirizzo [frasca@di.unimi.it](mailto:frasca@di.unimi.it) entro la data sopra indicata.

**Occorre presentare:**

1. **(Facoltativo)** Il codice sorgente C, adeguatamente commentato. I commenti alle singole righe vanno inseriti solo quando strettamente necessario. Mentre sono **obbligatori** i commenti all'inizio delle singole funzioni, specificando la logica della funzione, il significato di tutti gli argomenti e degli eventuali output.
2. **(Obbligatorio)** Una relazione in formato pdf che illustri in maniera esauriente la logica delle scelte implementative fatte per ogni operazione, l'algoritmo utilizzato (descrizione generale, **non commenti al codice**), analizzandone la complessità asintotica, e le strutture dati utilizzate per lo svolgimento delle singole operazioni.

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