

Tugas 1

Penggunaan ADT

Struktur Data dan Algoritma

IKI10400

Semester Gasal 2010/2011

Fakultas Ilmu Komputer

Universitas Indonesia

Batas waktu pengumpulan kode sumber:
Sabtu, 26 Maret 2011 pukul 21.00 Waktu Server Aren

Kode sumber yang dinilai hanya yang dikumpulkan melalui Aren.
Kode sumber yang dikumpulkan melalui mekanisme selain itu akan diabaikan dan dianggap tidak mengumpulkan.

Peringatan: jangan mengumpulkan pekerjaan beberapa menit menjelang batas waktu pengumpulan karena ada kemungkinan pengumpulan gagal dilakukan atau koneksi internet terputus!

Jika tidak dapat mengumpulkan tugas sebelum batas waktu karena suatu atau beberapa hal khusus, mahasiswa yang bersangkutan harus melakukan langkah-langkah dalam SOP Perpanjangan Batas Waktu Pengumpulan Pekerjaan.

FaceLang

Nama berkas kode sumber : SDA111111.java

Batas waktu eksekusi program : 3 detik / kasus uji

Batas memori program : 256 MiB / kasus uji

FaceLang adalah sebuah situs jejaring sosial yang berawal dari Universitas Harpord dan terus berkembang hingga terkenal di seluruh dunia. Setiap pengguna FaceLang memiliki **nama, daftar teman, daftar orang yang tidak disukai, daftar tempat wisata yang pernah dikunjungi, dan daftar hobi**. FaceLang memiliki fitur *Friend Suggestion* yang sangat unik karena fitur tersebut memiliki cara kerja yang sesuai dengan permintaan dari para pengguna FaceLang. Cara kerja dari fitur *Friend Suggestion* pada FaceLang adalah sebagai berikut.

- Pada setiap saat, seorang pengguna FaceLang hanya dapat melihat satu calon teman dari fitur *Friend Suggestion*.
- Untuk setiap pengguna FaceLang x , fitur *Friend Suggestion* tidak akan pernah menampilkan calon teman yang sebenarnya sudah berada di dalam daftar teman maupun daftar orang yang tidak disukai oleh x .
- Fitur *Friend Suggestion* hanya akan menambahkan maksimal satu calon teman pada setiap menit.
- Untuk setiap calon teman yang ditampilkan oleh fitur *Friend Suggestion*, dapat diberikan dua aksi yaitu:
 - *APPROVE*: keluarkan calon teman tersebut dari daftar pada fitur *Friend Suggestion* dan masukkan calon teman tersebut ke dalam daftar teman dari pengguna yang melakukan aksi ini.
 - *REJECT*: keluarkan calon teman tersebut dari daftar pada fitur *Friend Suggestion* dan masukkan calon teman tersebut ke dalam daftar orang yang tidak disukai dari pengguna yang melakukan aksi ini.
- Pengaturan prioritas yang menentukan calon teman yang akan ditampilkan fitur *Friend Suggestion* adalah sebagai berikut.
 - **Subprioritas 1.** Untuk setiap pengguna FaceLang x dan y , jumlah *mutual friend* antara x dan y didapatkan dengan cara menghitung banyaknya teman yang sama dari daftar teman x dan daftar teman y . Jika A dan B merupakan calon teman yang ditujukan ke C , dan jumlah *mutual friend* antara A dan C lebih besar dari jumlah *mutual friend* antara B dan C , maka A lebih diprioritaskan daripada B . Jika ternyata jumlah *mutual friend* sama, maka gunakan aturan subprioritas berikut ini.
 - **Subprioritas 2.** Untuk setiap pengguna FaceLang x dan y , jumlah *mutual tour* antara x dan y didapatkan dengan cara menghitung banyaknya tempat wisata yang sama dari daftar tempat wisata yang pernah dikunjungi x dan daftar tempat wisata yang pernah dikunjungi y . Jika A dan B merupakan calon teman yang ditujukan ke C , dan jumlah *mutual tour* antara A dan C lebih besar dari jumlah *mutual tour* antara B dan C , maka A lebih diprioritaskan daripada B . Jika ternyata jumlah *mutual tour* sama, maka gunakan aturan subprioritas berikut ini.

- **Subprioritas 3.** Untuk setiap pengguna FaceLang x dan y , jumlah *mutual hobby* antara x dan y didapatkan dengan cara menghitung banyaknya hobi yang sama dari daftar hobi x dan daftar hobi y . Jika A dan B merupakan calon teman yang ditujukan ke C , dan jumlah *mutual hobby* antara A dan C lebih besar dari jumlah *mutual hobby* antara B dan C , maka A lebih diprioritaskan daripada B . Jika ternyata jumlah *mutual hobby* sama, maka gunakan aturan subprioritas berikut ini.
- **Subprioritas 4.** Jika A dan B merupakan calon teman yang ditujukan ke C , s adalah waktu ketika A dimasukkan ke daftar *Friend Suggestion*, t adalah waktu ketika B dimasukkan ke daftar *Friend Suggestion*, dan s lebih dahulu daripada t , maka A lebih diprioritaskan daripada B .

Suatu ketika di Fakultas Ilmu Komputer UI (Universitas Innovatip), seorang mahasiswa jenius bernama Romeo menjadi yakin bahwa Nia adalah pasangan hidup yang dinantinya selama ini. Setelah tahu bahwa Nia merupakan pengguna baru FaceLang, Romeo mencoba melakukan PDKT dengan cara yang cerdas: dia berusaha mencari informasi sebanyak mungkin mengenai Nia dengan merangkum semua profil dari calon teman pada fitur *Friend Suggestion* FaceLang yang ditujukan ke Nia, dengan harapan rangkuman tersebut dapat menjadi senjata ampuh untuk mengenal Nia lebih dalam. Sebelumnya, Romeo telah membuat sebuah program canggih bernama MacLang Filter. Jika diberikan masukan seorang calon teman dari *Friend Suggestion* FaceLang, program ini akan menelusuri Internet dan menghasilkan sebuah profil lengkap mengenai orang tersebut¹. Program ini membutuhkan waktu selama 3 (tiga) menit untuk membangun profil untuk seorang calon teman.

Suatu hari, Romeo berhasil meng-*hack* FaceLang sehingga ia dapat menyadap daftar *Friend Suggestion* yang ditujukan ke Nia dalam kurun waktu 1x24 jam, dimulai pukul 00:00 agar tidak terdeteksi oleh pakar keamanan FaceLang. Kebetulan pada kurun waktu tersebut, Nia sedang liburan ke Puncak sehingga tidak mengakses FaceLang sama sekali. Selama 24 jam tersebut, Romeo menggunakan MacLang Filter untuk memproses daftar *Friend Suggestion* yang berhasil disadapnya. Setelah 24 jam tersebut berakhir, ia akan langsung bertemu Nia sepulangnya dari Puncak dan langsung mengajaknya ngobrol, tentunya dengan bekal profil yang sudah dibangun.

Ternyata daftar *Friend Suggestion* untuk Nia besar sekali sehingga MacLang Filter tidak dapat mengolah semuanya dalam 24 jam tersebut (ingat: FaceLang dapat menambahkan 1 calon teman per menit, sedangkan MacLang Filter hanya dapat mengolah 1 calon teman setiap 3 menit). Oleh karena itu, Romeo memikirkan bagaimana caranya agar MacLang Filter dapat bekerja dengan lebih efisien. Akhirnya Romeo mendapatkan ide bahwa tidak semua calon teman dari *Friend Suggestion* perlu diproses. Idennya adalah bahwa calon teman yang kemungkinan di-*REJECT* oleh Nia akan memperlambat proses dan merusak hasil rangkuman MacLang Filter. Oleh karena itu, MacLang Filter dimodifikasi sehingga hanya akan memproses seorang calon teman jika jumlah teman Nia yang tidak menyukainya kurang dari atau sama dengan $\alpha/2$, di mana α adalah jumlah teman Nia.

¹ Program ini akan menelusuri FaceLang, Twuter, MaiSpes, Prenster, SIAK-NG, dan masih banyak situs lagi!

Romeo meminta Anda membantu menyelesaikan MacLang Filter agar cita-citanya dapat tercapai.

Informasi Pada Masukan

1. Informasi pengguna direpresentasikan dengan format 5 baris sebagai berikut.

```
<Nama pengguna>
<Daftar teman-teman si pengguna>
<Daftar orang yang tidak disukai si pengguna>
<Daftar tempat wisata yang pernah dikunjungi si pengguna>
<Daftar hobi si pengguna>
```

Semua informasi yang berupa *<Daftar ...>* dapat berupa baris kosong, terdiri dari satu unit (yang dimaksud unit bisa berupa nama orang, nama tempat, atau nama hobi), atau terdiri lebih dari satu unit. Jika ada lebih dari satu unit dalam sebuah *<Daftar ...>*, tiap unit dipisahkan oleh sebuah karakter koma dan unit-unit itu dijamin unik dalam sebuah *<Daftar ...>*. Banyak teman si pengguna dijamin tidak lebih dari 3000. Banyak orang yang tidak disukai si pengguna dijamin tidak lebih dari 100. Banyak tempat wisata yang pernah dikunjungi ditambah banyak hobi si pengguna dijamin tidak lebih dari 20.

Karakter-karakter penyusun tiap unit dalam *<Daftar ...>* dan *<Nama pengguna>* hanya terdiri dari huruf kecil a..z dengan panjang minimal 1 karakter dan maksimal 6 karakter. Dijamin tidak akan ada nama yang sama:

- antara *<Nama pengguna>* dengan nama-nama di *<Daftar teman-teman si pengguna>*.
- antara *<Nama pengguna>* dengan nama-nama di *<Daftar orang yang tidak disukai si pengguna>*.
- antara nama-nama di *<Daftar teman-teman si pengguna>* dengan nama-nama di *<Daftar orang yang tidak disukai si pengguna>*.

2. Informasi calon teman direpresentasikan dengan format sebagai berikut.

```
<Waktu dimasukkan ke Friend Suggestion>
<Informasi pengguna yang merupakan calon teman Nia>
```

<Waktu dimasukkan ke Friend Suggestion> memiliki format "*<HH>:<MM>*" (untuk selanjutnya tanda kutip hanya untuk kejelasan) di mana $0 \leq HH < 24$ dan $0 \leq MM < 60$. *<Informasi pengguna yang merupakan calon teman Nia>* memiliki format sama seperti Informasi pengguna.

Format Masukan

Masukan dibaca dari masukan standar. Masukan diakhiri dengan EOF. Masukan dibagi menjadi 3 bagian yang tidak dipisahkan baris kosong atau pun baris yang berisi tulisan, yaitu:

- Bagian 1: informasi dari pengguna bernama Nia yang memiliki format sama seperti Informasi pengguna.
- Bagian 2: daftar informasi dari pengguna yang merupakan teman-teman Nia. Jika Nia belum memiliki teman, maka Bagian 2 tidak ada dalam masukan (bukan direpresentasikan dengan baris kosong). Setiap informasi tersebut memiliki format sama seperti Informasi pengguna. Daftar ini memiliki ukuran sesuai dengan banyak teman Nia pada Bagian 1. Misalkan Nia memiliki m teman, maka formatnya adalah sebagai berikut.

```
<Informasi teman Nia ke-1>
<Informasi teman Nia ke-2>
.
.
.
<Informasi teman Nia ke-m>
```

- Bagian 3: daftar informasi dari calon teman yang akan dimasukkan ke dalam Friend Suggestion. Jika tidak ada calon teman yang akan dimasukkan ke Friend Suggestion, maka Bagian 3 tidak ada dalam masukan (bukan direpresentasikan dengan baris kosong). Setiap informasi tersebut memiliki format sama seperti Informasi calon teman. Dijamin informasi-informasi pada Bagian 3 terurut menaik berdasarkan *<Waktu dimasukkan ke Friend Suggestion>*. Misalkan calon teman yang akan dimasukkan ke Friend Suggestion sebanyak n , maka formatnya adalah sebagai berikut.

```
<Informasi calon teman ke-1>
<Informasi calon teman ke-2>
.
.
.
<Informasi calon teman ke-n>
```

Dijamin setiap *<Nama pengguna>* yang terdapat pada ketiga bagian itu unik (misalnya jika *<Nama pengguna>* pada Bagian 1 adalah "nia", maka tidak akan ada teman Nia pada Bagian 2 yang bernama "nia"; begitu juga dengan Bagian 3). Dijamin pula unit yang sama merepresentasikan obyek yang sama (misalnya jika unit berupa nama orang adalah "ido" dan muncul beberapa kali pada masukan, antara "ido" yang satu dengan "ido" yang lain merepresentasikan orang yang sama). Dijamin m merupakan sebuah bilangan bulat nonnegatif yang tidak lebih dari 3000. Dijamin n merupakan sebuah bilangan bulat nonnegatif yang tidak lebih dari 3000.

Format Keluaran

Keluaran ditulis ke keluaran standar dengan format sebagai berikut.

```
<Daftar calon teman yang berhasil dirangkum MacLang Filter>
<Daftar calon teman yang tidak dirangkum MacLang Filter>
```

Setiap *<Daftar ...>* pada keluaran hanya mencantumkan nama dari calon teman yang

bersangkutan. Jika <Daftar ...> tidak terdapat nama, maka tampilkan "[KOSONG]". Jika pada sebuah <Daftar ...> terdapat lebih dari satu nama, maka pisahkan dengan menyisipkan sebuah karakter koma di antara tiap nama. Nama-nama dalam <Daftar calon teman yang berhasil dirangkum MacLang Filter> diurutkan berdasarkan urutan perangkumannya. Sedangkan dalam <Daftar calon teman yang tidak dirangkum MacLang Filter>, nama diurutkan secara leksikografis naik (*ascending*).

Contoh Masukan

nia anto riki tegal,bali makan,minum,tidur,main	Bagian 1
anto budi,denvil ido padang,merak golf,tidur	Bagian 2
18:22 henry budi,denvil,desy ido jepara,padang main	Bagian 3
18:23 ivon budi,denvil ido cina,india main	
18:25 jason budi,denvil ido kupang makan,minum,tidur	
19:00 ido anto,budi,denvil david bogor,depok makan,minum,tidur,main	

Contoh Keluaran

henry,jason,ivon ido

Penjelasan Contoh

Pada pukul 18:22, hanya ada calon teman bernama henry sehingga yang ditampilkan *Friend Suggestion* adalah henry. Semua teman Nia pada saat itu tidak ada yang tidak suka dengan henry sehingga MacLang Filter merangkum profil henry.

Ketika proses perangkuman pertama berakhir, yakni pukul 18:25, di dalam daftar *Friend Suggestion* telah masuk ivon pada pukul 18:23 dan jason pada pukul 18:25 sehingga perlu dilakukan perhitungan prioritas pada semua anggota yang ada di dalam daftar *Friend Suggestion*. Dengan hasil dari kalkulasi Subprioritas ke-3, didapatkan jason lebih diprioritaskan daripada ivon sehingga jason yang ditampilkan pada *Friend Suggestion*. Semua teman Nia pada saat itu (hanya anto) tidak ada yang tidak suka dengan jason sehingga MacLang Filter merangkum profil jason.

Ketika proses perangkuman kedua berakhir, yakni pukul 18:28, di dalam daftar *Friend Suggestion* tidak ada yang ditambahkan sehingga hanya tersisa ivon. Semua teman Nia pada saat itu (hanya anto) tidak ada yang tidak suka dengan ivon sehingga MacLang Filter merangkum profil ivon.

Ketika proses perangkuman ketiga berakhir, yakni pukul 18:31, tidak ada anggota di dalam daftar *Friend Suggestion*.

Hingga akhirnya pada pukul 19:00, ido masuk ke daftar *Friend Suggestion* dan ditampilkan. Akan tetapi, karena banyaknya teman Nia adalah 1 dan banyaknya teman Nia yang tidak suka dengan ido adalah 1 yaitu anto, maka profil ido tidak dirangkum oleh MacLang Filter karena tidak memenuhi kriteria "jumlah teman Nia yang tidak menyukainya kurang dari atau sama dengan $\alpha/2$, di mana α adalah jumlah teman Nia".

Kriteria Penilaian

Terdapat tiga bagian penilaian, yaitu:

- penilaian penilai otomatis Aren (40%).
- penilaian *white-box review* (40%).
- penilaian dokumentasi dan gaya penulisan pemrograman (20%).

Komponen-komponen penilaian *white-box review* adalah sebagai berikut.

- Terdapat kelas terpisah yang mengimplementasikan baca masukan dan tulis keluaran.
- Terdapat kelas terpisah yang merepresentasikan pengguna dan calon pengguna FaceLang.

- Pemilihan ADT yang tepat untuk menyimpan dan mengakses data.
- Implementasi yang benar dari pembandingan (harus menggunakan Comparator).