



شبکه‌ی اجتماعی:

یک شبکه‌ی اجتماعی از تعدادی «کاربر» تشکیل می‌شود، هر کاربر {نام، سن، جنسیت} دارد. در این شبکه‌ی اجتماعی، «صفحاتی» وجود دارد که هر «کاربر» می‌تواند به آن «صفحه» اظهار «علاقه‌مندی» کند. هر «کاربر» می‌تواند «کاربران» دیگر را «دنبال» کند (دو کاربر می‌توانند یک‌دیگر را دنبال کنند).

اطلاعات کاربران و صفحه‌ها در تعدادی فایل قرار گرفته‌است. تعداد کاربران و صفحه‌ها حداکثر ۲۰۰۰۰۰ نفر است که در حداکثر ۶۰۰۰۰ فایل نگهداری می‌شود و سایز هر فایل حداکثر ۱۵ مگابایت است. اطلاعات هر کاربر به صورت زیر مشخص می‌شود:

```
Start User
FirstName : Bruce;
SureName : Wayne;
Age : 38;
Gender : Male;
Slug : batman;
Followers : alfred, gordon, nightw, oracle, fox12;
IntrestedPages : jloa, bcave
OtherData : #####;
End
```

اطلاعات صفحه‌ها نیز به صورت زیر است:

```
Start Page
Title: Justice Leauge
Slug : jloa;
OtherData : #####;
End
```

فیلد slug به ازای هر کاربر و صفحه یکتا است و از حروف کوچک الفبای انگلیسی و اعداد تشکیل شده‌است. طول هر slug حداکثر ۶ کاراکتر است. در فیلد OtherData. ۲۰ کاراکتر خالی وجود دارد که می‌توانید برای اضافه‌کردن اطلاعات از این فضا استفاده کنید.

وظیفه‌ی شما پیاده‌سازی ساختاری برای نگهداری اطلاعات کاربران و همچنین ایجاد عملی برای تسریع در برخی عملیات در این شبکه‌ی اجتماعی است.

پایگاه‌داده:

برای نگهداری اطلاعات و دسترسی به آنها ساختار داده‌ی B+Tree را پیاده‌سازی کنید. برای کلید از slug استفاده کنید و value به شماره‌ی فایل و شماره‌ی کاراکتر شروع محدوده‌ی آن slug اشاره می‌کند. پس از پیاده‌سازی باید دو نمونه از B+Tree برای نگهداری صفحه‌ها و کاربران به صورت مجزا وجود داشته باشد.

وظایف عامل:

1- تسریع پایگاه‌داده برای موضوعات داغ: همان‌طور که احتمالاً دیده‌اید اتفاقات و اخبار مختلف به وسیله‌ی شبکه‌های اجتماعی به سرعت منتقل می‌شود. زمانی که یک موضوع خاص در صدر قرار دارد اکثریت یک شبکه راجع به آن موضوع صحبت می‌کنند، برای مثال می‌توان به مسابقات جام جهانی، انتخابات و مراسم اسکار اشاره کرد. در نتیجه برای کارایی

بیشتر پایگاه داده (با توجه به این که پایگاه داده در دیسک سخت قرار دارد) می‌خواهیم از ساختمان داده‌ای (در حافظه‌ی رم) استفاده کنیم تا در صورتی که موضوع داغ است برای جستجو به ساختمان داده‌ی B+Tree مراجعه نکنیم. برای پیاده‌سازی این قسمت، باید از ساختمان داده‌ی Treap استفاده کنید. کلید BST، همان کلید B+Tree و کلید Heap، دلخواه است. با توجه به این که حجم حافظه‌ی رم محدود است، ساختمان داده‌ی Treap نباید بیش از ۱۰۰۰ راس داشته باشد.

2- پیشنهاد دادن دوستی کاربرهای هم‌سلیقه به یک کاربر: دو نفر با هم امکان دوستی دارند اگر نزدیکی سلیقه‌ای آنها از عدد ثابت b کم‌تر باشد. نزدیکی سلیقه‌ای بین u و دیگر کاربران به این صورت تعریف می‌شود (مجموعه‌ی علاقه‌مندی یک نفر I_i است): «گرافی را در نظر می‌گیریم که به ازای هر کاربر یک راس وجود دارد و بین کاربر p و کاربر q یال وجود دارد اگر و فقط اگر حداقل یکی از آن‌ها دیگری را دنبال کند. وزن یال بین کاربر p و q برابر $\frac{1}{|I_u \cap I_q \cap I_p|}$ است. اگر طول کوتاه‌ترین مسیر وزنی از u به v برابر d بگیریم، نزدیکی سلیقه‌ای بین u و v برابر $\frac{|I_u \cap I_v|}{d}$ است.» برای پیاده‌سازی این قسمت به ازای هر کاربر، ۳ کاربر دیگر که امکان دوستی دارند و بیش‌ترین نزدیکی سلیقه‌ای به آن کاربر را دارند باید پیدا شوند. با توجه به این که این شبکه‌ی اجتماعی از تعداد خیلی زیادی کاربر تشکیل شده است الزامی ندارد ۳ کاربر با بیش‌ترین امتیاز، خروجی داده شود. در نتیجه شما می‌توانید برای سریع‌تر شدن زمان اجراء از تخمین نیز استفاده کنید. کارایی این قسمت به صورت تابعی از نزدیکی به جواب درست و زمان مصرف‌شده برای محاسبه است. در پایان در قسمت اطلاعات شخصی کاربر مورد نظر (که در یک فایل قرار دارد) slug این ۳ نفر باید اضافه شود.

3- تقسیم‌بندی کاربران به دو جامعه: کاربران در شبکه‌های اجتماعی عقاید متفاوتی دارند، که این عقاید باعث نزدیکی کاربران به یک‌دیگر و تشکیل جوامع (Communities) می‌شود. می‌خواهیم در این شبکه‌ی اجتماعی، کاربران را به دو گروه تقسیم کنیم که تعداد ارتباطات (follow میان دو گروه) کم‌ترین حد خود باشد (در این قسمت followها را دو طرفه در نظر بگیرید). برای پیاده‌سازی این قسمت یک الگوریتم تصادفی را پیاده‌سازی می‌کنید. در هر گام یک یال را به طور دلخواه انتخاب کنید، دو سر این یال را با هم در یک راس ادغام کنید و مشخصات رئوس ادغامی را در راس جدید بنویسید. حال یک راس خواهید داشت که همسایه‌های آن، همسایه‌های دو راس محذوف هستند. این کار را تا جایی که تنها یک یال باقی بماند انجام دهید. در این حالت هر راس یک جامعه در نظر گرفته می‌شود که کاربران نوشته شده در آن راس اعضای آن جامعه هستند. تعداد یال بین این دو جامعه برابر با تعداد یال بین هر دو عضو از دو جامعه است. احتمال درستی این الگوریتم حداقل برابر با $\frac{1}{\binom{n}{2}}$ است، پس برای به دست آوردن جواب درست با احتمال بالا، این الگوریتم را حداقل $\binom{n}{2}$ بار انجام دهید. در پایان در قسمت اطلاعات شخصی کاربر مورد نظر (که در یک فایل قرار دارد) شماره‌ی جامعه‌ی کاربر را بنویسید (این شماره ۱ یا ۲ است).

4- پیدا کردن بانفوذترین کاربران: می‌خواهیم در این شبکه‌ی اجتماعی یک خبر را پخش کنیم، کاربران می‌توانند نوشته‌ای را به اشتراک بگذارند و تمامی کسانی که آن کاربر را دنبال می‌کنند آن نوشته را می‌بینند و می‌توانند آن را پخش کنند. حال می‌خواهیم به تعدادی (کمینه) از کاربران خبری را اطلاع دهیم و به وسیله‌ی انتشار، تمامی کاربران از این خبر آگاه شوند. الگوریتمی پیاده‌سازی کنید تا این خواسته را برآورده کند. در قسمت اطلاعات شخصی این کاربران، بانفوذ بودن آن‌ها را مشخص کنید.

5- (امتیازی) به دست آوردن slug کاربر: در واقعیت، جستجوی دیگر کاربران به وسیله‌ی slug صورت نمی‌گیرد و از نام و نام خانوادگی استفاده می‌شود. ساختاری پیاده‌سازی کنید تا امکان به دست آوردن slug از روی نام یا نام خانوادگی

کاربر را فراهم کند. این قسمت نباید از حافظه‌ی RAM استفاده کند و در زمان کوتاهی باید نتیجه بدهد. در صورتی که چندین عضو با این نام یا نام خانوادگی وجود داشتند ۵ نفر اول که بیشترین مشابهت را دارند چاپ کند.

دستورات ورودی:

- 1- "init": این دستور پایگاه‌داده را مقداردهی اولیه می‌کند و در صورت اتمام پیغام "Build Successful." را چاپ می‌کند. در حین این دستور می‌توانید هرگونه محاسبه‌ای (که حتی به B+Tree ارتباطی ندارند) انجام دهید.
- 2- "suggest \$slug": این دستور برای کاربر با مشخص‌کننده‌ی \$slug، ۳ دوست پیشنهاد می‌کند و در اطلاعات آن کاربر می‌نویسد. هیچ خروجی در این قسمت نباید چاپ شود.
- 3- "makeCommunities": این دستور کاربران را به دو جامعه تقسیم می‌کند و برای هر کاربر در قسمت اطلاعات، شماره‌ی گروه را می‌نویسد. در این قسمت تعداد لینک‌های بین دو جامعه باید چاپ شود.
- 4- "findMostInfluenceUsers": این دستور افراد با نفوذ را شناسایی می‌کند و تعداد آن‌ها را چاپ می‌کند.
- 5- "page \$slug": این دستور عنوان صفحه‌ای با مشخص‌کننده‌ی \$slug را خروجی می‌دهد.
- 6- "profile \$slug [options]": این دستور مشخصات کاربر با مشخص‌کننده‌ی \$slug را خروجی می‌دهد. در صورتی که هیچ گزینه‌ای در ورودی نباشد، نام و نام خانوادگی کاربر و دنبال‌کننده‌های او را در دو خط به صورت زیر می‌نویسد. برای مثال:

```
>profile batman
```

```
Name: Bruce Wayne
```

```
Followers : Alfred Pennyworth, James Gordon, Dick Grayson, Barbara Gordon, Lucius Fox
```

Options:

- "-n": تنها نام و نام خانوادگی کاربر نوشته می‌شود.
- "-s": نام و نام خانوادگی ۳ کاربری که به این کاربر پیشنهاد داده می‌شود.
- "-c": شماره‌ی جامعه‌ای که کاربر به آن اختصاص داده شده‌است.
- "-f": نام افرادی که این کاربر را دنبال می‌کنند.

نکات پایانی:

- 1- فایل‌ها در یک پوشه به نام «Data» کنار فایل‌های اجرایی برنامه قرار می‌گیرد. همچنین شما می‌توانید اطلاعات اضافی خود را در یک پوشه به نام «MyData» نگهداری کنید حجم این پوشه نباید بیش از ۵۰ مگابایت شود.
- 2- یال‌های گراف در قسمت ۲ و ۳ بدون جهت هستند.
- 3- برای آزمودن قسمت‌های ۳ و ۴ از حداکثر ۱۰۰ کاربر استفاده می‌شود.
- 4- استاندارد ورودی و خروجی باید مشابه استانداردهای گفته شده باشد.
- 5- این تمرین در قالب تیم‌های حداکثر دو نفره انجام می‌شود.
- 6- در صورت هر گونه تقلب و مشابهت کدها (حتی در اینترنت) نمره‌ی این تمرین برابر با 100- در نظر گرفته می‌شود. با توجه به نمره‌ی بالای این تمرین لطفاً به این نکته دقت بیشتری کنید.
- 7- این تمرین تحویل حضوری دارد. هر دو نفر گروه باید به تمامی قسمت‌های کد مسلط باشند و بتوانند از تئوری هر قسمت دفاع کنند.

- 8- برای این تمرین از کامپایلر Java 1.7 استفاده کنید. در ضمن تحویل این تمرین در سیستم عامل ubuntu خواهد بود.
- 9- ایده‌های مطرح شده را (حتی اگر به مرحله‌ی پیاده‌سازی نرسند) در قسمت ۱ (برای کلید Heap) و قسمت ۲ (تابع تخمینی) در فایل Ideas.pdf بنویسید (ایده‌هایی که در کد وجود دارند را مشخص کنید)
- 10- فایل‌های برنامه‌های خود را می‌توانید در پوشه‌های مختلف نگه‌داری کنید. فایل Main.java در بیرونی‌ترین پوشه باشد و تابع main در این فایل پیاده‌سازی شود. یک فایل Makefile در بیرونی‌ترین پوشه بگذارید که به وسیله‌ی آن تمامی پروژه کامپایل شود و Main.class در بیرونی‌ترین پوشه قرار بگیرد.
- 11- فایل‌های پروژه را به همراه Makefile و Ideas.pdf را در CA7DS[SID1-SID2].zip قرار داده و بارگزاری کنید.

شاد باشید و امیدوار