



پاییز ۱۴۰۱

پروژه پایانی درس ساختمان داده‌ها

❖ شرح پروژه

میخواهیم یک سیستم کتابخانه راه اندازی کنیم که شامل تعدادی کتاب و تعدادی عضو است. هر عضو میتواند به کتابخانه مراجعه کرده و تعدادی کتاب را به مسئول کتابخانه سفارش دهد.

مسئول کتابخانه اسم کتاب را درون سیستم وارد می‌کند و در صورتی که از آن موجود بود مشخص می‌کند که درون کدام ردیف و قفسه است و به مسئول آوردن کتاب آن را می‌سپارد. مسئول آوردن کتاب بعد از یافتن کتاب آن را بر میدارد و اگر متقاضی هنوز در کتابخانه باشد آنرا می‌آورد.

هر عضو در زمان ورود به سیستم ورود و خروج در ورودی نام خود را میدهد و سیستم ورود او را ثبت می‌کند. هر عضو پس از ورود کتاب‌های که با خود برگردانده را به مسئول پذیرش کتاب‌ها می‌دهد و او وظیفه دارد بازگشت این کتاب و زمان آن را در سیستم ثبت نماید.

مسئول در زمان خروج هر عضو باید کتاب‌های که قرض گرفته و نام آن عضو را در سیستم ثبت نماید. سپس هر عضو در زمان خروج باید نام خود را در سیستم ورود و خروج ثبت نماید و خارج شود.

مسئول کتابخانه می‌تواند کتاب‌هایی که اهدا شده اند و یا جدیداً به کتابخانه آورده شده اند را در سیستم ثبت نماید.



پاییز ۱۴۰۱

❖ می‌خواهیم این سیستم را با تابع‌های زیر و پیچیدگی زمانی بهینه که برای هر کدام ذکر شده است پیاده کنیم (تمامی اسم‌ها Unique هستند):

1) arrive(personName)

این تابع ورود اعضا را به سیستم ثبت می‌کند.

پیچیدگی این تابع از $O(\text{len}(\text{personName}))$ است.

2) exit(personName)

این تابع خروج اعضا را به سیستم ثبت می‌کند.

پیچیدگی این تابع از $O(\text{len}(\text{personName}))$ است.

3) isInLib(personName)

این تابع نام شخص را ورودی گرفته و مشخص می‌کند که او همچنان در کتابخانه است

یا خارج شده است.

پیچیدگی این تابع از $O(\text{len}(\text{personName}))$ است.

4) borrowBook(personName, bookName)

این تابع می‌گوید که کتاب با نام bookName توسط personName قرض گرفته شد

پیچیدگی این تابع از $O(\text{len}(\text{personName}) + \text{len}(\text{bookName}))$ است.

5) returnBook(bookName, personName)



پاییز ۱۴۰۱

این تابع می‌گوید که کتاب با نام `bookName` توسط `personName` بازگردانده شد.

پیچیدگی این تابع از $O(\text{len}(\text{personName}) + \text{len}(\text{bookName}))$ است.

6) TotalTimeInLib(personName, startTime, endTime)

این تابع یک نام و یک بازه زمانی می‌گیرد و مجموع تعداد ساعت‌های که این فرد در

این باز زمانی در کتابخانه بود است را خروجی می‌دهد.

پیچیدگی این تابع از $O(\text{len}(\text{personName}) + \log(\text{Number of entry and exit}))$ است.

7) addNewBook(bookName, count)

این تابع به تعداد `count`، کتاب با اسم `bookName` در سیستم اضافه می‌کند

پیچیدگی این تابع از $O(\text{len}(\text{bookName}))$ است.

8) shouldBring(bookName, personName)

این تابع می‌بیند که اگر فرد متقاضی با اسم `personName` هنوز در کتابخانه بود و کتاب با اسم `bookName` نیز موجود بود آن را از قفسه مربوطه آورده و به وی می‌دهد.

پیچیدگی این تابع از $O(\text{len}(\text{personName}) + \text{len}(\text{bookName}))$ است.

9) allPersonCurrentBooks(personName)

این تابع تمام کتاب‌های حال حاضر که توسط شخص با نام `personName` گرفته شده است را مرتب شده به صورت الفبای برمیگرداند.

پیچیدگی این تابع از $O(\text{len}(\text{personName}) + (\text{Number of borrowed books}))$ است.

10) allPersonsHaveThisBook(bookName)



پاییز ۱۴۰۱

این تابع نام تمام افرادی که این کتاب را در حال حاضر قرض گرفته اند را به ترتیب الفبای برمیگرداند.

پیچیدگی این تابع از $O(len(bookName) + (Number\ of\ Persons\ borrowed\ this\ book))$ است.

11) allPersonsBorrowedThisBook(bookName)

این تابع نام تمام افرادی که این کتاب را در کل قرض گرفته اند را به ترتیب الفبای برمیگرداند.

پیچیدگی این تابع از $O(len(bookName))$ است.

12) allPersonAllTimeBooks(personName)

این تابع تعداد کتاب‌هایی که تا به حال توسط شخص با نام `personName` گرفته شده است برمیگرداند.

پیچیدگی این تابع از $O(len(personName))$ است.

نکات

- پروژه انفرادی می‌باشد.
- تمامی کدها با استفاده از نرم افزار مشابه گیر MOSS بررسی خواهند شد. هیچ دو پروژه‌ای با یکدیگر نباید بیشتر از ۱۰ درصد شباهت داشته باشند. (در صورت وجود شباهت، این موضوع به استاد اطلاع رسانی خواهد شد و به دانشجویان اطلاع رسانی خواهد شد.)
- کد غیر بهینه نمره‌ی بالایی دریافت نخواهد کرد.
- برای انجام پروژه می‌توانید از زبان‌های ++C و Java استفاده کنید.
- در زبان ++C تنها استفاده از کتابخانه‌های `iostream` و `string` مجاز می‌باشد.
- در زبان Java تنها استفاده از کتابخانه‌های `java.util.Scanner` مجاز می‌باشد.
- دقت کنید که Exception‌ها نباید روند اجرای برنامه شما را مختل کنند.
- زمانبندی تحویل حضوری متعاقبا اعلام خواهد شد.
- در زمان تحویل حضوری ممکن است از شما خواسته شود تغییراتی در کد اعمال کنید.