

به نام خدا

تمرین عملی سری سوم درس ساختمان داده

نکته: این تمرین شامل یک پروژه گروهی است که ۳۵ نمره و دو تمرین که ۱۵ نمره دارند، در مجموع تمرین عملی شماره سه ۵۰ نمره دارد.

نکته: پروژه به صورت گروهی و گروه‌ها دو نفره است، پس از ایجاد یک رپوزیتوری در گیت‌هاب توسط حل التمرین عملی درس، شروع به انجام مرحله به مرحله پروژه کنید، این مراحل را در فایل `readme.md` توضیح دهید و مشخص کنید چه عضوی از گروه کدام بخش از پروژه را انجام داده است.

نکته: سعی کنید `commit` های مکرر داشته باشید به صورتی که بعد از انجام هر بخش یک `commit` انجام شده باشد، این کار مشخص میکند پروژه در چند روز و در چند مرحله توسط شما انجام شده و کدام بخش‌ها را چه عضوی از گروه نوشته است.

نکته: قبل از شروع تمرین انتظار می‌رود ساختمان داده‌های آرایه پویا، آرایه ایستا، صف، پشته، لینک لیست یک‌طرفه و دوطرفه، درخت جست‌وجوی دودویی تدریس شده باشد، با توجه به این موضوع لطفاً سعی کنید از ساختمان داده‌هایی مانند درخت برای، هرم‌ها، جدول درهم سازی و.. استفاده نکنید.

نکته: هر کدام از دو تمرین داده شده را در یک فایل به صورت جداگانه پیاده‌سازی کنید. نام این فایل‌ها را به ترتیب `t1.py` و `t2.py` قرار دهید.

18 صورت پروژه از قرار زیر است، پروژه به گونه‌ای تنظیم شده که ظرف دو هفته قابل
انجام باشد:

20 یک کاربر عادی به شما به عنوان یک برنامه ساز که در حوزه ساختمان داده
22 تخصص دارد ایمیلی ارسال کرده، در این ایمیل از شما خواسته شده که برنامه‌ای برای
24 استفاده شخصی او طراحی و پیاده‌سازی کنید، لطفاً در ابتدا راهکارها، ساختمان داده‌ها،
روش حل مسأله خود را بیان و سپس این برنامه را بنویسید. این نرم‌افزار جهت استفاده
شخصی مشتری شما است، این برنامه جهت مدیریت فایل‌های شخصی کاربر استفاده
میشود، فایل‌هایی که قصد اضافه کردن آن‌ها به کامپیوتر دارد را ابتدا در این برنامه
26 ثبت میکند، برنامه پس از دریافت نام فایل، اندازه بر حسب مگابایت و محل ذخیره
سازی آن را در سیستم ثبت میکند. یک حداکثر اندازه حافظه کامپیوتر در تنظیمات
28 برنامه وجود دارد که میتوان آنرا در هر لحظه تغییر داد. در صورتی که حجم مجموع
فایل‌های موجود در سیستم از حداکثر حافظه موجود بیشتر شود باید قدیمی ترین
30 فایل‌ها به صورت خودکار حذف شوند، این اتفاق در دو حالت زیر رخ می‌دهد:

1. اندازه حداکثر حافظه به مقدار کمتری تغییر داده شود و مجموع حجم فایل‌ها از
32 آن بیشتر باشد.

2. قصد افزودن فایل جدید داشته باشیم اما قبل از اینکه فایل را اضافه کنیم با
34 اندازه‌گیری حجم مجموع بعد از افزودن، متوجه شویم که اندازه از حداکثر بیشتر
خواهد بود.

36 پس از اینکه در هر کدام از حالات فوق مشخص شود که با کمبود حافظه مواجه
هستیم، فایل‌های قدیمی را به ترتیب قدیمیترین حذف میکنیم تا اینکه حافظه برای
38 انجام هر کدام از دو عملیات فوق (افزودن فایل جدید و یا مقدار جدید اندازه کل
حافظه) کافی باشد.

40 همچنین مشتری می‌خواهد لیستی از فایل‌های موجود در سیستم را به ترتیب
بزرگترین به کوچکترین و یا بالعکس، به ترتیب آخرین فایل‌های اضافه شده و یا بالعکس
42 مشاهده کند.

پنل کاربری:

44 پس از باز شدن برنامه گزینه‌های زیر با چینی دلخواه و البته کاربر پسند باید در
اختیار باشد:

- 46 ۱. یک کنسول که اعمال برنامه در آن چاپ می‌شود، در این کنسول عملیات‌های
افزودن فایل، حذف فایل، برای هر فایل نام و اندازه‌اش و زمان چاپ شود.
- 48 ۲. یک گزینه برای تغییر دادن حداکثر اندازه حافظه سیستم وجود داشته باشد.
- 50 ۳. یک گزینه برای افزودن فایل جدید به سیستم، در این حالت نام، اندازه، آدرس
حافظه فایل دریافت می‌گردد و سپس فایل به سیستم اضافه می‌شود.
- 52 ۴. لیستی از فایل‌های موجود در سیستم که در حالت پیشفرض بر اساس
جدیدترین فایل‌ها تنظیم شده، این لیست را میتوان با یک گزینه در هر کدام از
چهار حالت مرتب سازی تنظیم نمود.
- 54 ۵. یک بخش که به صورت آنلاین و بروزرسانی لحظه‌ای میزان فضای اشغالی،
باقی‌مانده و کل را نشان می‌دهد.
- 56 پس از بسته شدن برنامه تمامی اطلاعات را در هارد کامپیوتر ذخیره کنید و در
لحظه شروع آن داده‌ها را بارگزاری کنید. هزینه زمانی عملیات‌های مربوط به این بخش
58 اهمیت چندانی ندارد، میتوانید در فایل‌های متنی و باینری داده‌ها را ذخیره کنید. به
هیچ عنوان در حین کار برنامه از فایل‌های درون هارد استفاده نکنید. (فقط در لحظه
60 شروع و پایان)

تمرین ۱. برای انجام این تمرین، ابتدا تمرین شماره دو تئوری را انجام دهید، سپس با توجه پیاده‌سازی زیر، متد `route` را به گونه‌ای بنویسید که بتواند دو گره (ند) `u` و `v` را دریافت کند، سپس مسیر ساده‌ای بین این دو گره را پیدا کند و همه گره‌های درون این مسیر را در قالب یک `list` پایتون خروجی دهد. (۱۰ نمره)

```
class BinaryTree:
    class Node:
        def __init__(self, data):
            self.data = data
            self.parent = None
            self.left = None
            self.right = None

        def __init__(self):
            self.root = None

        def route(self, u, v):
            """
            Finds a way from u(Node type) to v(Node type) and
            returns the nodes in way.
            :param u: First node
            :param v: Second node
            :return: a list containing all nodes from u to v
            """
```

برای پر کردن این درخت با داده‌های آزمایش می‌توانید از تکه کدی مشابه زیر استفاده کنید و یا اینکه متدهای مربوطه را بنویسید:

```
(x = BinaryTree
x1 = BinaryTree.Node("x1")
x.root = x1
x2 = BinaryTree.Node("x2")
x1.left = x2
x2.parent = x1
x3 = BinaryTree.Node("x3")
x1.right = x3
x3.parent = x1
```

تمرین ۲. تمرین شماره شش در بخش تئوری را پیاده‌سازی کنید. (۵ نمره)

94 `def find_closest(root: Node, k: int) → int:`
 `pass`

96 موفق باشید.