



دانشگاه گیلان

پروژه سوم ساختمان داده و الگوریتم

دکتر میرروشندل

ترم ۱۴۰۲۱

گیلان بوک



هدف

هدف از انجام این پروژه آشنایی دانشجویان با ذخیره داده ها در درخت و آشنایی با درخت پیشنهاد (suggest tree) و استفاده از این ساختمان داده در محیط عملی میباشد.

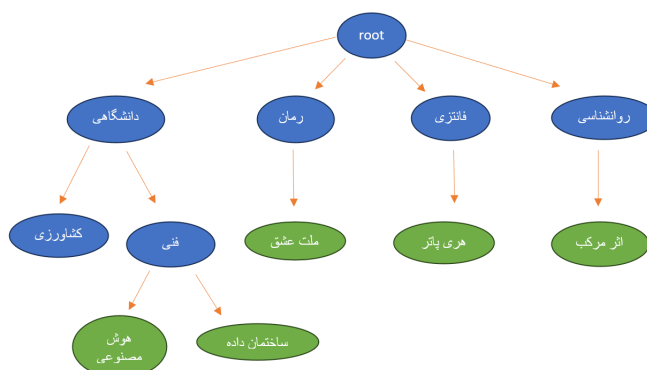
مقدمه

پس از اینکه دانشجویان نتوانستند کتاب هایی که در ترم یک خریده بودند را بفروشند، (رو دستشون باد کرد!) از دانشگاه تقاضا کردند که سامانه ای مناسب برای این کار داشته باشند. این موضوع به گروه مهندسی کامپیوتر اطلاع داده شده و از شانس خوب شما، این مسئله با پروژه سوم درس ساختمان داده همزمان شده! دکتر میروشندل نیز از فرصت استفاده کرده و به عنوان ارزشیابی مستمر، طراحی این سامانه را به عهده شما سپرده که با یک تیر دو نشان زده باشد!

شما به عنوان یکی از دانشجویان نخبه دانشگاه گیلان وظیفه دارید سامانه ای جدید برای دانشگاه گیلان طراحی کنید که در آن دانشجویان کتاب های دسته دوم خود را میفروشند.

توضیح

درخت و دردرسرهایش: باید فضایی ایجاد شود که کاربر بتواند هر کتاب را در دسته بندی مناسب خودش به فروش بگذارند. یا در دسته بندی مورد علاقه خودش به دنبال کتاب مورد نظر خود بگردد یا از طریق نام آن را جستجو کند.



autocomplete حرفه ای: هنگام جستجو، برنامه باید بتواند نام کتاب هایی که مشابهت اسمی با نام وارد شده توسط کاربر دارند را به او پیشنهاد دهد. این عمل باید با ساختمان داده درخت پیشنهادی پیاده سازی شود.



تکمیل خودکار (Autocomplete)

برای این بخش باید از ساختمان داده درخت پیشوندی استفاده کنید. اطلاعات بیشتر:

<https://en.wikipedia.org/wiki/Trie>

*** راهنمایی های بیشتر در کلاس حل تمرین ***

از Ternary Search Tree هم مجاز هستید که استفاده کنید، اما درخت پسوندی مفهومی ساده تر دارد و پیشنهاد میشود. برای مطالعه و تمرین بیشتر، بد نیست نیم نگاهی به این داده ساختار نیز داشته باشید.

امکاناتی که برنامه شما باید داشته باشد:

۱- اضافه کردن دسته بندی A به دسته بندی های B: دسته بندی A به دسته بندی B اضافه میشود. با متوذهای مربوطه دسته بندی B را سرچ کنید و دسته بندی را به آن اضافه کنید. حتما اطمینان حاصل کنید که اولاً دسته بندی ای با نام A در درخت وجود نداشته باشد، و دوماً دسته بندی ای با نام B وجود داشته باشد.

۲- حذف دسته بندی A: دسته بندی A سرچ میشود و به همراه تمامی کتاب هایش حذف میشود.

۳- اضافه کردن کتاب A به دسته بندی B: کتابی شامل مشخصات (نام کتاب، نویسنده کتاب، قیمت کتاب، دسته بندی کتاب) اضافه میشود. با متوذهای مربوطه دسته بندی B را سرچ کنید و کتاب را به آن اضافه کنید. حتما اطمینان حاصل کنید که اولاً کتابی با نام A در درخت وجود نداشته باشد، و دوماً دسته بندی ای با نام B وجود داشته باشد.

۴- حذف کتاب A: کتاب A سرچ و حذف میشود.

۵- نمایش کتاب های یک دسته بندی بر اساس جستجوی کتگوری: تمام کتاب های دسته بندی و زیردسته بندی های آن را نمایش دهید.

۶- نمایش یک کتاب بر اساس جستجوی نام و درخت trie: کاربر ممکن است اسم کتاب را کامل وارد نکند. تمام کتاب هایی که با رشته وارد شده شروع میشوند را نمایش دهید.

۷- پنل مشخصات کتاب: پس از جستجو در مرحله ۵ و ۶، با وارد کردن نام دقیق کتاب، تمام اطلاعات مربوط به آن کتاب اعم از نام، قیمت، نویسنده و... را نمایش دهید. دقت کنید این امکان تنها پس از جستجو در ۵ و ۶ باید فراهم باشد.

روابط شیء‌گرایی عناصر برنامه، از جمله کتاب، دسته بندی، و... باید صحیح و اصولی پیاده سازی شوند.

محدودیت هایی که برنامه شما باید رعایت کند:

۱- اینکه چگونه درخت را پیاده سازی کنید دست شماست، اما در تعداد نودهای درخت ها نباید هیچ محدودیتی باشد. برای نگهداری نودهای فرزند هر نود میتوانید از LinkedList یا آرایه پویا استفاده کنید. حواستان باشد از هیچ ساختار آماده ای استفاده نکنید.

۲- نام کتاب ها و دسته بندی ها منحصر به فرد است، برنامه نباید اجازه دهد هیچ کتاب یا دسته بندی ای دپلیکت شود.

راهنمایی‌ها و ساده‌سازی‌های پروژه

- نام کتاب‌ها و دسته‌بندی‌ها، فقط شامل حروف انگلیسی و کارکتر whitespace است. عدد ندارد.
- بهتر است نود کتاب و نود دسته‌بندی از یکدیگر متمایز باشند. هر دو از نود درخت extend کنند و خواص منحصر به فرد هر یک، جداگانه پیاده‌سازی شود.
- حواستان باشد هنگام نمایش کتاب‌ها، دسته‌بندی‌ها را نمایش ندهید، بلکه **فقط کتاب‌های آن دسته‌بندی و زیردسته‌بندی‌هایش را نمایش دهید.**
- یک دسته‌بندی می‌تواند **هم شامل کتاب و هم شامل دسته‌بندی‌های دیگر باشد.**
- نیازی به ایجاد رابط کاربری جداگانه‌ای برای فروشنده و خریدار نیست. می‌توانید تصور کنید که کتابخانه را یک ادمین مدیریت می‌کند و فقط همین شخص دسترسی به برنامه و تمام امکانات آن را دارد.
- نیاز به پنل ثبت نام نیست.
- حتماً راه‌های مناسبی جهت پیمایش گزینه‌های برنامه قرار دهید. یعنی برنامه‌گزینه‌هایی مثل go back داشته باشد.
- Bad Input ها را هندل کرده و با ارور مناسب به کاربر اطلاع دهید.

امتیازی

- با استفاده از درخت پسوندی تکمیل خودکار علاوه بر اینکه از اول انجام شود، از آخر نیز انجام شود. مثلاً با ثبت کلمه عشق، کلمات زیر پیشنهاد شوند: عشق نامرئی، عشق لرزه، ملت عشق، زهر عشق
- اضافه کردن امکانات جدید با استفاده از داده‌ساختارهای متفاوت مثل Binary Search tree و ... (از پروژه‌های قبلی هم می‌توانید استفاده کنید)

ثبت پروژه

پروژه خود را در فایل فشرده Zip با فرمت زیر بنویسید و آن را در کوئرا ارسال کنید.

DSProject۳_FirstnameLastname_StudentNumber.zip

نکات عمومی پروژه‌ها

هرگونه ایده و خلاقیت جدید، تنها با به کارگیری مباحث پروژه و سرفصل‌های مرتبط با این درس و با هماهنگی با دستیاران آموزشی، تا ۲۵ درصد نمره مثبت به همراه خواهد داشت.

پروژه شما تنها در صورتی مشمول **نمره مثبت** میشود، که بخش اصلی پروژه را **۹۰ درصد** تکمیل کرده باشید.

رابط گرافیکی **نمره مثبت ندارد.**

استفاده از هرزبان برنامه‌نویسی **بلامانع است.**

کپی و استفاده کورکورانه از منابع اینترنتی، یا ابزارهایی نظیر Chat-GPT **ممنوع است**.

پروژه تنها به صورت انفرادی قابل انجام بوده و وجود هرگونه تشابه مشکوک بین دو کد یا عدم تسلط به ارائه، **شامل جریمه خواهد بود**.

تمامی ساختمان داده‌های مورد نیاز پروژه باید توسط شما پیاده سازی شوند و استفاده از ساختمان داده‌های موجود در کتابخانه‌های زبان برنامه نویسی مورد نظر شما **شامل کسر نمره خواهد بود**.

تنها داده ساختار مجاز **آرایه ساده** است. `ArrayList, Vector` و ... **نامجاز هستند و جهت استفاده باید خودتان آنها را پیاده سازی کنید**.

با ساختار List در پایتون **فقط مثل آرایه ساده رفتار کنید**. از امکاناتی که مربوط به آرایه ساده نیستند، استفاده نکنید. (مثل پوش و پاپ) از **دیکشنری پایتون استفاده نکنید**.

رعایت اصول شیء گرایی و آموزه‌های درس برنامه‌سازی پیشرفته، نام گذاری صحیح، کامنت نویسی مناسب، رعایت دندانه گذاری، و خوانا بودن کد **الزامی است**.