

پروژه SQL فایناول  
(سوال - مقدماتی)

محمد رضا خواجه‌پوری

عملیات	Array collection	LinkedList collection	توضیح
find By ID	$O(n)$	$O(n)$	در <del>push back</del> Array لیست سریع در هر دو ساختار
find All	$O(n)$	$O(n)$	کل لیست را باید بار و اسکن کنیم
count/sum/avg	$O(n)$	$O(n)$	باید بهای تمام شده کامل داده
delete One	$O(n)$	$O(n)$	باید کل لیست را بررسی کنیم تا ID پیدا شده بعد حذف کنیم
insert one	$O(1)$	$O(1)$	در <del>push back</del> Array در <del>node</del> linked list با اضافه کردن جدید هر دو ثابت هستند

لیدر

تذاری نیست عملکرد

فای این پروژه برای تست جدید رکورد دانشجو رو از فایل data.txt می‌خواند و مدل ذخیره سازی یعنی Array collection و LinkedList collection امتحان کردم روی ~~توی~~ هر دو ساختار

جداگانه مثل اضافه کردن دانشجو (insert one) پیدا کردن با آیدی (find By ID) در فای  
همه دانشجو (find All) و محاسبه تعداد و معدل (average, count) و محاسبه کردن و رفتار  
برنامه رو بررسی کردم نتیجه این بود که برای همین تعداد کم داده هر دو ساختار خوب و بدون مشکل کار  
می‌کنند ولی اضافه کردن رکورد جدید فای Array collection کمی روان تر و سریع تر انجام می‌دهد  
از طرفی در عملیات فای مثل find All و محاسبه کردن تعداد و معدل همان اجرا در هر دو ساختار  
تقریباً یکی است و تفاوت خاصی دیده نمی‌شود فقط اگر تعداد داده ما خیلی زیاد شود انتظار می‌رود  
این تفاوت بیشتر خود ساختار رو فاش می‌کند

قسمت clean code

برای این قسمت روی پروژه سعی شده که یک پروژه مرتب باشد برای مثال اسم کلاس ها مثل student و Query Parser با حرف بزرگ و اسم متغیر و توابع مثل studentList به این صورت انتخاب شده تا ظاهر یک دست داشته باشد به واسطه که نوشتن تعریف شده و دوباره سازی Array Collection و LinkListCollection از روی آون ساده تر شده تا بدون دست زدن به بقیه برنامه ها نوشته بشه رفع ذخیره سازی و عوضی کردن

که ها به چندین بخش ولایه تقسیم شده اند که هر بخش کار خودش را انجام داده و متعلق ها قاطبی نشده مثل storage , engine , parser , execution

حالا بیایم که نظری برده به جای یکی کردن کد روی توابع و کلاس های مشترک جمع شده تا حجم کار کمتر بشه هم نگهداریش راحت تر هست