گزارش پروژه پایانی
ساختمان داده و الگوریتم

:نام و نام خانوادگی:امیررضا خان محمدی
:شماره دانشجویی4023110412
:تاریخ تحویل

ساختار نرم افزار

ساختار فایل‌های نرم افزار

```
FinalProject/
├── main.py
├── models/                 #[Type:Folder]
│   ├── Car.py                    #[Type:python class]
│   ├── City.py         #[Type:python class]
│   ├── FineNode.py      #[Type:python class]
│   ├── HistoryPlate.py  #[Type:python class]
│   ├── Plate.py         #[Type:python class]
│   ├── User.py          #[Type:python class]
├── Menu/               #[Type:Folder]
│   ├── Login.py         #[Type:python File]
│   ├── ManagerPanel.py  #[Type:python File]
│   ├── Signup.py        #[Type:python File]
│   ├── UserPanel.py
├── DataBase/           #[Type:Folder]
│   ├── CarData.py       #[Type:python File]
│   ├── CarHitory.py     #[Type:python File]
│   ├── CityNumber.py           #[Type:python File]
│   ├── FineHistory.py     #[Type:python File]
│   ├── PlateHistory.py    #[Type:python File]
│   ├── Userinfo.py       #[Type:python File]

├── Data_structure/             #[Type:Folder]
│   ├── structure.py             #[Type:python class]

├── TestFile/             #[Type:Folder]
```

```
│   ├── Car.txt                # [Type: Text File]
│   ├── citycode.txt           # [Type: Text File]
│   ├── ownership_history.txt # [Type: Text File]
│   ├── penalties.txt          # [Type: Text File]
│   ├── phase4.txt             # [Type: Text File]
│   ├── users.txt              # [Type: Text File]
```

کلاس ها

[Array] کلاس شماره1:
یک ارایه به طول ثابت به ما میده که برای دخیره سازی کد هر شهر و اسم شهر و پلاک های ان استفاده شده(پلاک ها در درخت دودیی قرار دارند)

Main method: insert , slice_array

[Hash] کلاس شماره2:

یک هش میسازه و واسه هندل کردن کالیژن هم لینر استفاده میکنه و برای دخیره کردن اصلاعات هر فرد و برای ذخیره کردن جریمه استفاده میشود

Main method: insert , search , delete , incerase_hash ,changekey

[Linklist] کلاس شماره3:

لینک لیست دوطرف هست هم برای ذخیره کردن هر پلاک مخصوص به فرد که برای ان است (در اصل در نودی که برای هش میسازم یک لینک لیست هم وجود دارد که واسه پلاک های ان فرد هست)

Main method: insert , Linsert(insert to last) , delete , search ,

[BST] کلاس شماره 4:

برای ذخیره سازی ماشین ها که کلید ان سریال ماشین است و برای ذخیره سازی پلاک ها هست که کلید ان هم خود پلاک است

Main method : insert , search , findmin(use when we want delete) , delete , in_order (for print BST)

[DynamicArray] کلاس شماره 4 :

برای ذخیره کردن جریمه ها و تاریخچه ماشین که در هر نود یوزر و پلاک استفاده میشود و در فاز 4 هم برای ذخیرر امتیاز ها ستفاده میشود

Main method : insert , increasearray , restart array(array will be None)

[MakePlate]کلاس شماره 5:

برای ساختن پلاک ها است و شرایط که در پلاک ها وجود داشت برای هر پلاک و تکراری نبودن ان را بررسی میکند

Main method : Platemake , checkplate

[CarNode] کلاس شماره 6:

نود که برای ماشین ها تعرف شده و شامل اسم و سریال و پلاک و رنگ و کد ملی مالک و تاریخ ساخت و داناميک ارایه برای ذخیره خرید فروش ان

[CityNode] کلاس شماره 7:

نود که برای هر شهر ساخته میشه شامل کد شهر اسم شهر و درخت دودیی برای ذخیره پلاک هر شهر

[Fine] کلاس شماره 8:

نود برای جریمه که شامل تاریخ جریمه و درجه جریمه و توضیحات و سریال راننده پلاک مربوط به جریمه است

[PlateHistory] کلاس شماره 9:

نود برای عوض شدن پلاک روی ماشین ها که شامل سریال ماشین کد ملی مالک پلاک شروع مالکیت و پایان مالکیت

[PlateNode]کلاس شماره 10 :

نود برای ذخیره پلاک ها که شامل شهر ان کد ملی مالک سریال ماشین وضعیت پلاک و ارایه برای ذخیره جریمه ها و یک ارایه دیگه برای خرید فروش ماشین

[User] کلاس شماره 11:

نود برای ذخیره اطلاعات کاربر که شامل کد ملی نام نام خانوادگی رمز عبور یک لینک لیست برای پلاک که مربوط ان وضعیت راننده بودن ان سریال ان اگر راننده بود نمره منفی روز هایی که نمیتواند پلاک بگیرد و ارایه برای جریمه انها

ساختمان های داده و الگویتم‌ها

ساختمان های داده

Every data_struture in cd/Data_structure/structure

[Array] ساختمان داده شماره1:

در بعضی از جاها یک سایزه داده رو میدونستیم و ثابت است مثل نام شهر ها و کد ان ها استفاده میکنم

راحتی سرچ در ان بدون نوشتم الگوریتم خاصی در ان به راحتی سرچ کرد و ساده بودن در پیاده سازی

عملیات هم اینزرت ساده دارد و اسلایس ارایه که برای پلاک ها ساخته میشود و حرف رو میان ان قرار بدهیم

[Hash] ساختمان داده شماره 2:

دسترسی به داده به هزینه 1 انجام میشود همینطور سرچ برای دخیره کردن کاربر ها استفاده میشود که کد ملی انها باید تا باشد و در هش کلید تکراری قبول نمیکند و چون تعداد کاربر ها رو نمیدونیم هش میتوانند تا بی نهایت داده در خود ذخیره کند (هش داینامیک هست) عملیات اینزرت سرچ کردن عوض کردن کلید

[Dlllinklist] ساختمان داده شماره3:

سایز ان ثابت نیست و مشخص نیست هر نفر چند پلاک دارد و چرا از ارایه استفاده نکرد چون نسبت لینک لیست حافظه خیلی بیشتری مصرف میکند عملیات هام شامل اینزرت دیلیت و سرچ هست

[BST] ساختمان داده شماره 4:

جستجو بهینه تر نسبت به ارایه و فضای اشغالی کمتر نسبت به هش و قابلیت حذف بدون جابه جا کردن بقیه عناصر عملیات شامل اینزرت حذف سرچ کردن و پیمایش ان

الگوریتم ها

[Quick-sort] الگوریتم شماره1:

Cd/Finalproject/Faze4.py

برای مرتب سازی راننده بر اساس زمان گرفتن گواهینامه خود

پیچدگی زمانی : O(nlogn)

پیچدگی مکانی : O(د)

پیاده‌سازی نرم‌افزار

ساختمان داده‌های اصلی:

[Hash_Data] ساختمان داده شماره 1:

Cd/DataBase/Userinfo.py

نود های مربوط به هر کاربر در خود دارد که هرنود شامل نام و نام و خانوادگی و تاریخ تولد و رمز عبور ان و لینک لیستی که شامل پلاک های انها باشد وضعیت راننده بودن ان اگر راننده هم بود تاریخ گواهی نامه و نمره منفی تاریچه جریمه ان

[Hash_Fine] ساختمان داده شماره 2:

Cd/DataBase/FineHistory.py

نود های مربوط به جریمه راه نگه میدارد که هر نود شامل تاریخ جریمه و سطح جریمه و توضیحات و کد پیگیری جریمه و پلاک و سریال راننده

[City_Array] ساختمان داده شماره 3:

Cd /DataBase/CityNumber.py

نود ها مربوط به هر شهر را داد که نود شامل اسم شهر کد شهر و یک درخت دودویی از پلاک ها مربوط به ان شهر و هر نود پلاک که در درخت است شامل کد ملی مالک و تاریخه خرید و فروش جریمه و نام شهر و اگر سریال ماشین هم داشت در ان وجود دارد و وضعیت فعالی ان

[CarBST] ساختمان داده شماره 4:

نود ها مربوط به هر ماشین را نگه میدارد که هر نود شامل رنگ نام سال تولید پلاک خوردو و کد ملی مالک و تاریخچه خرید و فروش

عملکردها

[ثبت نام کاربر]عملکرد 1:

Cd /Menu/Signup.py/function(Signup)

منطق و الگوریتم:

اصلاعات ار کاربر دریافت میکند و برای ان یک نوئ میسازد و در هش ان دخیره میکند

پیچیدگی:

پیچیدگی زمانی: O(1)

پیچیدگی مکانی: O(n) n تعداد کاربر

[ایجاد پلاک] عملکرد شماره 2:

Cd/models/plate.py/class MakePlate

منطق و الگوریتم:

با استفاده از کتاب خونه رندوم یک عدد 5 رقمی میسازد و با کتاب خونه استرینگ یک حرف به میان اضاف میکند

پیچیدگی:

[مشاهده خودرو های ثبت شده] عملکرد شماره 3:

Cd/Menu/UserPanel.py/function(SeeCar)

منطق و الگوریتم:

با کد ملی کابر در هش ان را پیدا کرده بعد از لینک لیست پلاک ها پلاک های ان که سریال ماشین دارن را سرچ میکنم

پیچیدگی:

پیچیدگی زمانی: O(logn*m) m تعداد ماشین ها

پیچیدگی مکانی: O(m)

[مشاهد پلاک های ثبت شده] عملکرد شماره4:

Cd/Menu/UserPanel.py/function(SeePlate)

منطق و الگوریتم:

کد ملی کاربر در هش سرچ میکند و از لینک لیست پلاک ها ان را چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(1)

پیچیدگی مکانی: O(v) v تعداد پلاک ها

[پلاک کردن خودرو] عملکرد شماره5:

Cd/Menu/ManagerPanel.py/Function(Car_registration)

منطق و الگوریتم:

مشخصات ماشین رو دریافت میکنه و درخت ماشین ها اضافه میکند

پیچیدگی:

پیچیدگی زمانی: O(logn)

پیچیدگی‌مکانی : O(1)

[مشاهده خودرو های ثبت شده] عملکرد شماره 6:

Cd/Menu/ManagerPanel.py/function(SeeAllCar)

منطق و الگوریتم:

درخت ماشین رو چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(n)

[مشاهد کابران ثبت شده] عملکرد شماره 7:

Cd/Menu/ManagerPanel.py/function(SeeALLUser)

منطق و الگوریتم:

هش کاربران رو چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(n)

[مشاهد پلاک های یک شهر] عملکرد 8:

Cd/Menu/ManagerPanel.py/function(SeeALLCityPlate)

منطق و الگوریتم:

اسم شهره دریافت میکنه و در ارایه شهر پلاک های اون رو چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(n)

[مشاهد ماشین های یک شهر] عملکرد شماره9:

Cd/Menu/ManagerPanel.py/function(SeeALLCarCity)

منطق و الگوریتم:

با دریافت اسم شهر و پیدا کردن درخت پلاک ها هر پلاکی که سریال ان وجود داشت ان را پرینت میکنه

پیچیدگی:

پیچیدگی زمانی: O(m*logn) M تعداد پلاک های یک شهره

پیچیدگی‌مکانی: O(n)

[جستووجوماشین ها در بازه زمانی] عملکرد شماره10:

Cd/Menu/ManagerPanel.py/function(DateSearch)

منطق و الگوریتم:

با توجو به بازه گرفته شده از ورودی کل درخت پیمایش میکنه و با توجه به تاریخ چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(n)

[مشاهد مالکان یک شهر] عملکرد شماره 11:

Cd/Menu/ManagerPanel.py/function(SeeALLOwnerCity)

منطق و الگوریتم:

نام شهر در ارایه شهر ها سرچ کرده و درخت پلاک را با پیمایش و کدملی ها را به هش داده و مالکان شهر را چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(n)

[تغییر نام کاربری] عملکرد شماره 12:

Cd/Menu/Manager.py/function(ChangeUsername)

منطق و الگوریتم:

کاربر را در هش پیدا میکند و اصلاعات ان را دخیره میکند و ان را حذف میکند و با نام کاربری جدید اضاف میکنه

پیچیدگی:

پیچیدگی زمانی: O(1)

پیچیدگی‌مکانی: O(1)

[مشاهد نمره منفی] عملکرد شماره 13:

Cd/Menu/UserPanel.py/function(User_Seenegative)

منطق و الگوریتم:

کاربر را در هش سرچ میکند و نمره منفی را پرینت میکند

پیچیدگی:

پیچیدگی زمانی: O(1)

پیچیدگی‌مکانی: O(1)

[مشاهد تاریچه جرایم] عملکرد شماره 14:

Cd/Menu/UserPanel.py/Fumction(SeeFine)

منطق و الگوریتم:

کاربر را در هش پیدا میکند و ارایه جرایم را چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(1)

پیچیدگی‌مکانی: O(1)

[مشاهد جرایم یک پلاک خاص] عملکرد شماره15:

Cd/Menu/UserPanel.py/Function(SeeFinePLate)

منطق و الگوریتم:

پلاک را دریافت و میکنه و کاربر رو هم در هش سرچ میکنه اگر پلاک متعلق به خوش بود تاریچه را نمایش می دهد

پیچیدگی:

پیچیدگی زمانی: O(1)

پیچیدگی‌مکانی: O(1)

[مشاهد تاریخچه پلاک] عملکرد شماره 16:

Cd/Menu/UserPanel.py/function(History)

منطق و الگوریتم:

کاربر را در هش پیدا میکند و اگر پلاک متعلق به او بود تاریخچه پلاک را چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(1)

پیچیدگی‌مکانی: O(1)

[مشاهد تاریخچه خرید و فروش خودرو] عملکرد شماره17:

Cd/Menu/ManagerPanel.py/function(SeeCarHistory)

منطق و الگوریتم:

با دریافت سریال ماشین ان را در درخت ماشین سرچ میکند و تاریخچه رو چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(logn)

پیچیدگی‌مکانی: O(1)

[مشاهد رانندگان ثبت شده] عملکرد شماره 18:

Cd/Menu/ManagerPanel.py/function(RegisteredDriver)

منطق و الگوریتم:

کل هش را پیمایش میکنه و هر کدوم که وضعیت رانندگی داشت را چاپ میکنئ

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(1)

[تغییر مالکیت شماره پلاک] عملکرد شماره 19:

Cd/Menu/ManagerPanel.py/function(chnageowner)

منطق و الگوریتم:

پلاک را پیدا میکند و سریال ماشین را بر میدارد و به پلاک جدید می دهد

پیچیدگی:

پیچیدگی زمانی: O(logn)

پیچیدگی‌مکانی: O(1)

[حذف خودرو] عملکرد شماره 20:

Cd/Menu/ManagerPanel.py/function(DeleteCar)

منطق و الگوریتم:

در درخت ماشین ماشین را پیدا میکند و ان را حذف میکند

پیچیدگی:

پیچیدگی زمانی: O(logn)

پیچیدگی‌مکانی: O(1)

[اعطای مجوزه رانندگی] عملکرد شماره21:

Cd/Menu/ManagerPanel.py/function(Drivinglicense)

منطق و الگوریتم:

کاربر در هش پیدا میکند و به ان مجوزه میدهد و وضعیت رانندگی فعال میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(1)

[حذف مجوز رانندگی] عملکرد شماره22:

Cd/Menu/ManagerPanel.py/function(removelicense)

منطق و الگوریتم:

راننده را هش پیدا میکند و اصلاعات ان را مربوط به رانندگی هست را حذف میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(1)

[تغییر حالت راننده] عملکرد شماره23:

Cd/Menu/ManagePanel.py/function(BlockingDriver)

منطق و الگوریتم:

رانند را پیدا میکند و وضعیت رانندگی را غیر فعال میکند

پیچیدگی:

پیچیدگی زمانی: O(n)

پیچیدگی‌مکانی: O(1)

[مشاهد تاریخچه مالکیت خودرو] عملکرد شماره 24:

Cd/Menu/ManagerPanel.py/function(SeeownerCarHistory)

منطق و الگوریتم:

ماشین را در درخت ماشین پیدا میکند و تاریخچه را چاپ میکند

پیچیدگی:

پیچیدگی زمانی: O(logn)

پیچیدگی‌مکانی: O(1)

[ثبت جریمه] عملکرد شماره 25:

Cd/Menu/ManagerPanel.py/function(MakeFine)

منطق و الگوریتم:

با دریافت کردن مشخصات جریمه در هش فاین ذخیره میکند و برای پلاک و ماشین ها ذخییره میکند

پیچیدگی:

پیچیدگی زمانی: O(logn)

پیچیدگی‌مکانی: O(1)

چالش ها

چالش هایی که در این پروژه با آن مواجه شدید؟

1-تغییرمالکیت پلاک که در کد پلاک عوض شده و هنوز به نام فرد باقی می ماند

2-جستو و جو بر حسب تاریخ ماشین که واسه ورودی گرفت به چه صورت باشد

3-تغییر نام کاربری که پس از ری هش درست کار نمیکرد

4- ساخت پلاک با توجه به شرایط که گفته شده بودن و نبودن پلاک تکراری

چالش هایی که در پیاده سازی با آن مواجه شدید؟

1-حذف کردن هش تیبل

2- حذف نود در بی اس اتی

3- ری هش کردن

نظر شما در مورد پروژه چیست؟

نظر خاصی ندارم

منابع

[https://www.geeksforgeeks.org]1-واسه پیدا سازی ساختمان داده ها

Chatgpt

[https://www.python.org/]