بسم الله الرحمن الرحیم

**طراحی شده توسط :** محمدمهدی مجریان

**شماره دانشجویی :** 40213260281815

برای نوشتن الگوریتم ها استفاده شد. vscode**محیط برنامه نویسی :** از برنامه

**زبان برنامه نویسی :** از زبان python برای نوشتن اسکریپت استفاده شد، به دلیل اینکه زبان پایتون دارای syntax ساده و دارای متد ها کاربردی زیادی هست که کار را برای نوشتن و درک کردن الگوریتم سریع و ساده تر می کند ، هرچند که در ساخت زبان پایتون از زبان C هم استفاده شده.....
اما سرعت بسیار کمتری نسبت به زبان C دارد ، ولی باز هم پایتون به عنوان یک زبان سطح بالا کارش درسته....

## توضیحاتی درمورد دستورات کد :

اگر بخوام مختصر مفید بگم ، در ریپاسیتوری ایجاد شده دوتا فایل پایتونی به نام های main.py و sorts.py ایجاد کرده ایم.

از فایل sorts.py برای نوشتن الگوریتم های مرتب سازی استفاده کرده ایم که ان را با استفاده از سه تا فانکشن (اصلی) insertation_sort و merge_sort و Hybrid_sort انجام دادیم که هرکدوم با الگوریتم مشخص خودشون لیست ما را مرتب سازی میکنن اگر بخوام یکم خلاصه درمودر توابع های ایجاد شده بگم :

Merge_sort : با استفاده از تقسیم کردن لیست ما به دوبخش با تابع بازگشتی تا جایی که به خونه های تک عضوری تبدیل شوند و سپس هر دو آرایه تک عضوی با هم مقایسه می شوند و یک ارایه دو تایی می شوند و این روند تا انجا ادامه می یباد

که به سایز ارایه اصلی اما مرتب شده برسیم پیچیدگی زمانی این الگوریتم در تمامی حالات nlogn هست.

Insertion_sort : این الگوریتم میاد از خونه دوم تا خونه آخر را تراول می کند و هر خونه را با خونه های قبل خودش مقایسه می کند و عملیات مرتب سازی را انجام می دهد و برای ارایه هایی با سایز کم بهینه تر است و به درد ارایه ها با سایز بزرگ نمی خورد و پیچیدگی زمانی O(n^2)را خواهد داشت در حالتی که خیلی خوش شانس باشیم و ارایه ما مرتب باشد میشه O(n)چون بازم باید n-1 تا مقایسه انجام بدی بازم

Hybrid_sort : همانطور که از اسمش مشخصه ترکیبی از این دوتاست اگر سایز kما کمتر از 100 باشد از الگوریم درجی استفاده می کند در غیر این صورت از ادغام استفاده می کند

**خروجی برنامه:** در نهایت برنامه زمان اجرا یک ارایه با اعداد رندونم و سایز رندوم برای مرتب کردنش را به ما نشان می دهد

ان چیزی که مشاهده شد این است که الگوریتم مرتب سازی درجی برای ارایه هایی با سایز بزرگ بهینه نیست و زمان خیلی بیشتری نسبت به دو الگورتیم دیگر دارد

مرتب سازی ادغامی و مرتب سازی دوگانه با اختلاف اندک از هم و تفاوت فاحش با الگوریتم مرتب سازی درجی بهینه تر هستند برای ارایه ها با سایز بزرگ ( هرچند که با اختلاف خیلی خیلی کم در بیشتر جا ها دوگانه کمی بهتر بود ، به صورت خیلی ناچیز)

در آخر هم از دستور assert برای اینکه مطمعن بشیم که هر سه تا الگوریتم ارایه را مرتب کردند استفاده کردیم تا به ما نتیجه فیک ندهد

نکته: از ماژول copy کمک گرفتیم چون که زمانی که یک لیست داشته باشیم و یک فانکشن ان را مرتب کند ان لیست دیگر برای بقیه فانکشن ها نیز مرتب شده و این موجب می شود که در نتیجه نهایی اشتباه کنیم برای همین از یک ارایه دو تا کپی دیگر گرفتیم تا نتیجه دقیق باشد ، باشد که رستگار شویم....