In the name of God



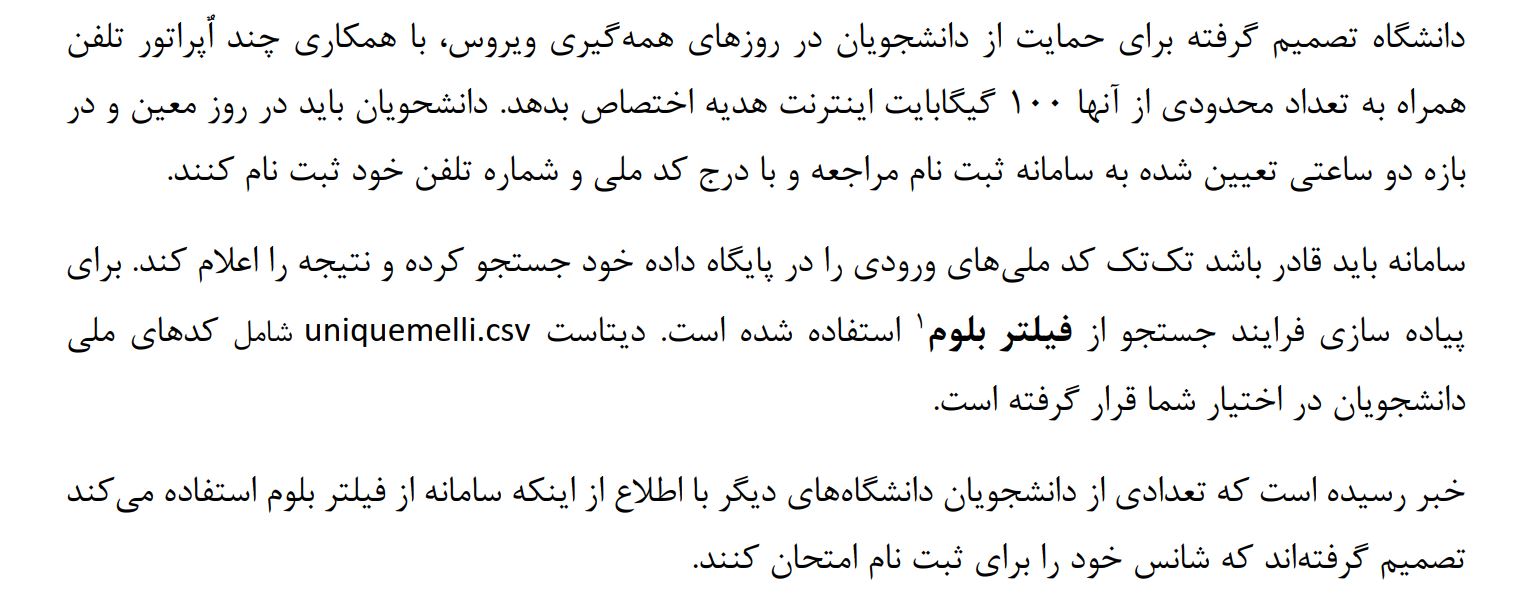
استاد : دکتر هراتی

دانشجو : توحید حقیقی سیس

شماره دانشجویی : 830598021

موضوع : تمرین سوم

**تمرین اول :**



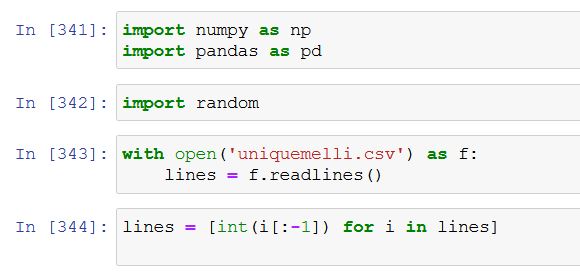
فیلتر بلوم به طور کلی به این صورت است که ما در ابتدا یک ارایه تمام صفر با طول دلخواه در نظر میگیریم و یه تعداد تابع هش در این تمرین هدف تعیین کردن تعداد تابع های هش مناسب و طول ارایه اولیه مناسب است

من این تمرین را با پایتون در محیط جوضیتر پیاده سازی کردم که مرحله به مرحله توضیح میدهم .

* پیاده سازی به صورت کلاینت سرور : در این قسمت من داده ها را تک تک در ارایه شامل یک ها تولید و چک میکنم تا ببینم این کد ملی در لیست دانش جویان دانشگاه تهران هست یا نه .

هر داده ای از سمت سرور گرفته شود اول هش شده و در ارایه اولیه چک میشود تا ببینیم به خانه ای هش میشود که 1 است اگر یک بود الگوریتم ما این فرد را به صورت اشتباه تشخیص داده و False positive رخ میدهد .

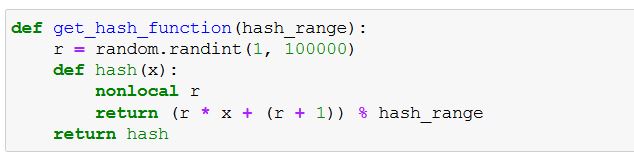
* برای مشاهده موارد بالا از ابتدا باید پیاده سازی را توضیح دهم .
* مرحله اول : خواندن داده ها از دیتاست و ساختن ارایه ای که شامل یک ها است .



در تابع بالا دیتاست موجود را خوانده و به صورت لیست اعداد نگه میدارم و خروجی این قسمت به صورت زیر است .

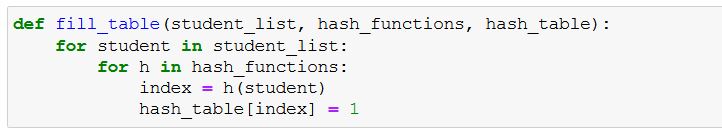


در قسمت بعد تولید ارایه از روی دیتاست موجود را با تاع fill\_table انجام میدهم اما برای اینکه این تابع را توضیح دهم قبل از ان باید تابع هش که در این تابع انجام شده را توضیح دهم .

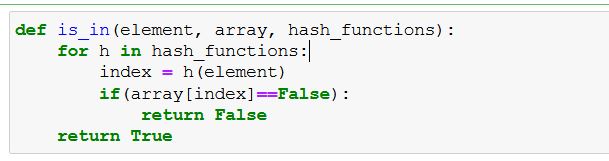


ما در این تابع یک عدد رندوم درست کرده و از روی ان تابع ax+b را تولید کرده و خود func را برمیگردانیم .

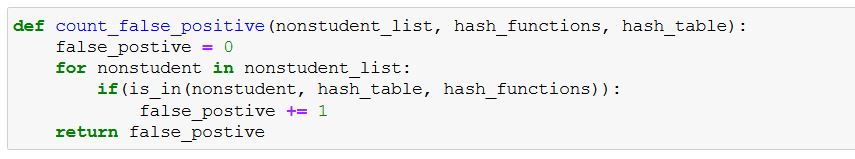
این قسمت برای این است که چون در روش بلوم فیلتر لیستی از هش ها لازم است که یک شماره ملی را با همه ان ها هش کرده و در ان خانه ها یک قرار دهیم و برای تشخیص و یا جستجو کد ملی نیز اول ان عدد را با تمام هش ها هش کرده اگر در همه ان ها به 1 رسید بس در ارایه قرار دارد .



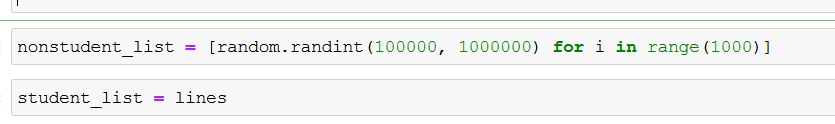
در تابع بالا ارایه را به وسیله تابع هش قسمت قبل پر میکنیم ولی یک مورد را باید دقت کنیم که هر چه قدر ارایه اولیه بیشتر باشد تعداد 1 های موجود در ان به نصبت کمتر میشود چون خانه های زیادی دارد و هش ها به خانه های متفاوت میتواند اعداد را هش کند و الگوریتم جستجو ما دقیق تر میشود اما باید زیاد هم بزرگ نباشد که بی دلیل حافظه بگیرد در قسمت اخر این سوال من با طول ارایه مختلف و تعداد هش های مختلف تست کردم و به صورت نموداری ان ها را نمایش داده ام .



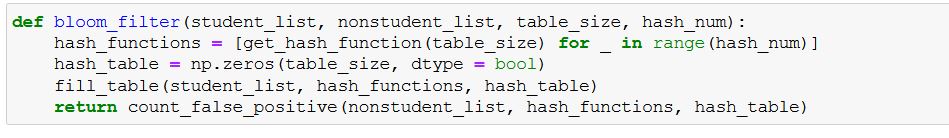
تابع بالا برای بررسی این که یک عدد که برای دانشجویان خارج از دانشگاه تهران گرفتیم به صوزت اشتباهی میتواند ثبت نام کند یا نه منظور از اشتباهی عدد رندومی که داده در تمام هش ها به یک خانه میوفتد یا نه .



در این سوال یک مفهوم False positive است یعنی کد ملی اختیاری که زدیم جزو لیست کد ملی ها نیست ولی به اشتباه روش ما ان را جزو ان کد ملی ها میگیرد .

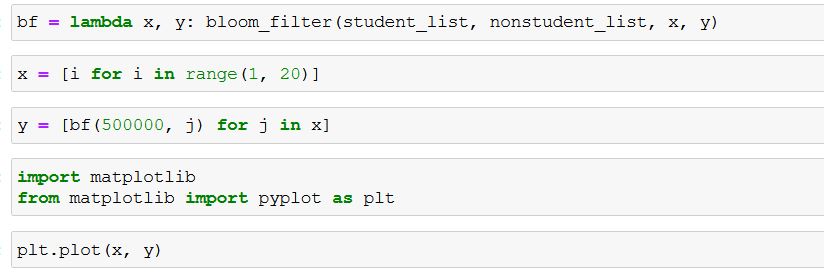


در تابع بالا لیست داده های رندوم که در زیر ایجاد کردیم بنابر صورت سوال 1000 تا ایجاد کردیم . الگوریتم ما ان کد ملی را میگه جزو لیست هست یا نه برای محاسبه ان با تمام هش ها ان را بررسی و باید در تمام هش ها به مقدار 1 برسد .



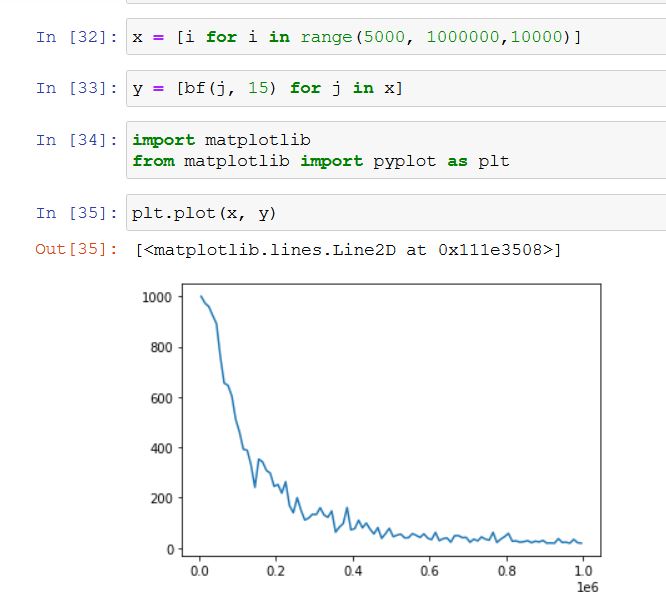
به صورت کلی تابع فیلتر بلوم من این مراحل را انجام میدهد و توابع بالا که توضیح دادم را فراخوانی میکند .

من توابع بالا رو بر اساس یکی از قسمت های سوال با تعداد هش متفاوت و طول ارایه متفاوت اجرا کردم و نمودار ان ها را در زیر قرار دادم .



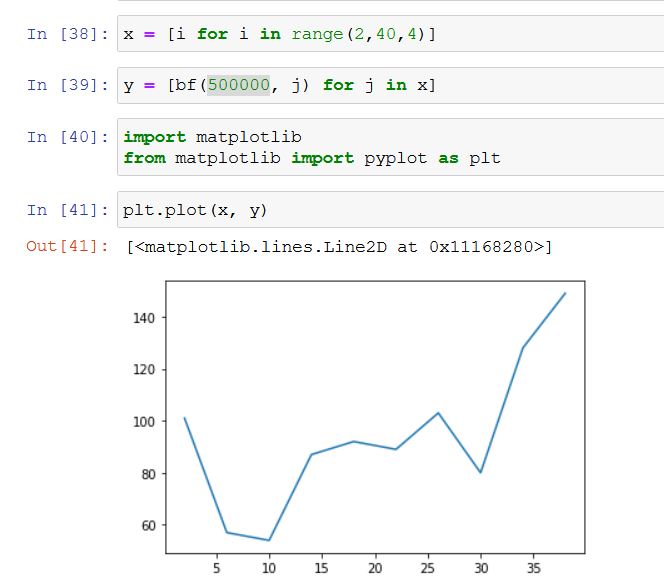
کد رسم نمودار به صورت بالا است و در زیر اجرا و خروجی ان را قرار میدهم .

قسمت اول با تعداد هش 15 طول ارایه را تغییر میدهم از 5000 تا 1000000 :



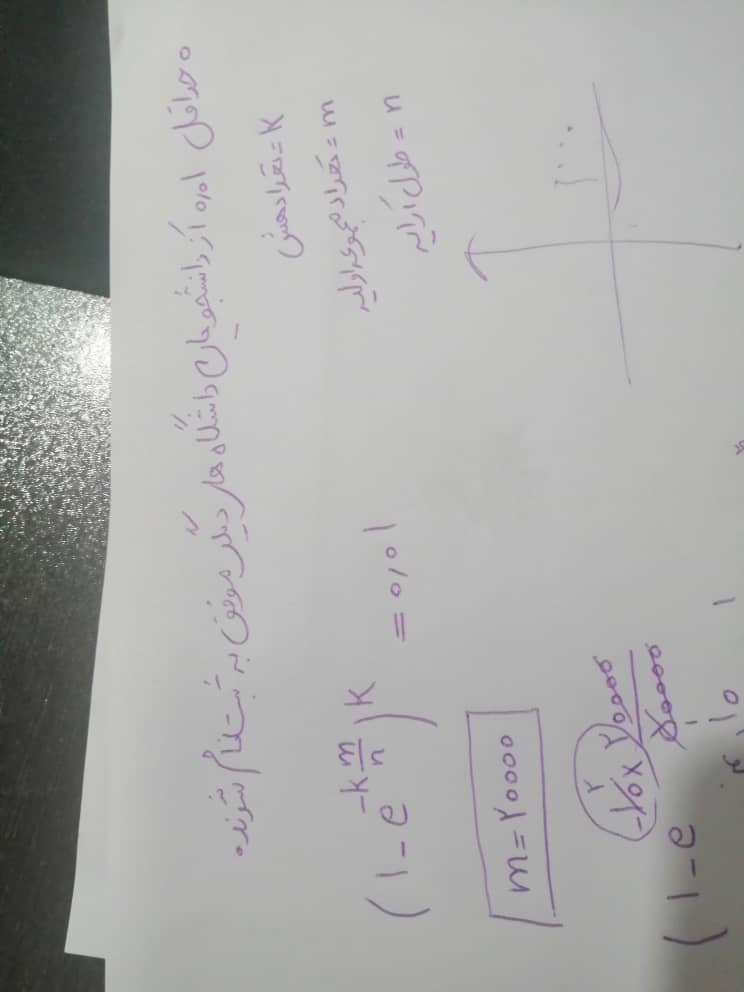
شکل بالا خیلی واضح نشان میدهد که هر چه طوا ارایه اولیه بیشتر شود تعداد افرادی که به صورت رندوم بخواهند وارد سیستم شوند کمتر میشود یعنی خطای false positive کمتر میشود .

قسمت دوم طول ارایه را ثابت میگیرم و تعداد هش ها را تغییر میدهم .



بر اساس شکل بالا هر چه تعداد هش بالا بره الگوریتم بهتر کار نمیکنه در اعدادی مثلا 10 در این مثال میبینیم که این عدد به کمتریم مقدار خود رسیده است یعنی false positive در کمترین مقدار خود است .

* ایا همه دانشجویان دانشگاه تهران موفق به ثبت نام میشوند ؟ جواب این سوال بله است چون این روش False negative ندارد یعنی افرادی که در دیتاست قراردارند و با هش های مختلف در جدول قرارگرفته اند اگر با همان شماره ملی دوباره بخواهند وارد شوند هش همان است و به راحتی میتوانند ثبت نام کنند بس جواب این سوال بله هست .
* برای اینکه 0.01 دانشجویان دانشگاه های دیگر بتوانند ثبت نام کنند .

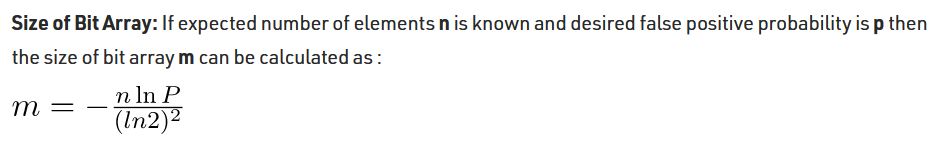
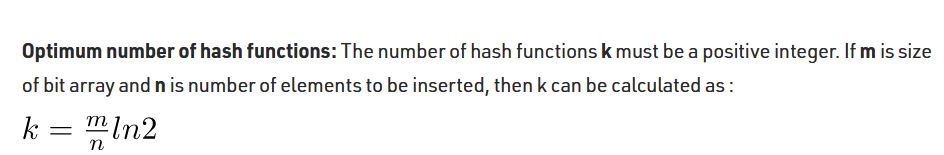


يبق شکل بالا باید حساب شود با چه اعدادی ان تساوی بر قرار است با محسابات من یکی از نقاطی که در ان صدق میکند .

تعداد هش : 10

طول ارایه : 400000

چند فرمول زیر نیز برای محاسبه k و n استفاده میشود :



که با محاسبه m=20000 در این صورت با فرمول پایین n=2070 به دست می آید.

و k = 10\*0.69=6.9 به دست می آید .

**تمرین دوم :**

****

برای حل این قسمت نیز از روش هش استفاده میکنیم به این صورت که به اعداد باینری هش میکنیم و تعداد صفر ها را از اخر شمارش میکنیم اگر تعداد صفر های ان برابر بود به یک نفر اشاره میکند .

برای هر دو دیتاست زیر این الگوریتم را اجرا میکنیم برای این که شمایه کلاینت – سروری بدهد اعداد را تک تک وارد و بزرگترین سول داده شده را نگه میداریم .

دیتاست اول :digikala.xlsx چون این دیتاست از نوع اکسل است به صورت زیر خوانده میشود .



کد بالا فایل اکسل را میخواند و در ws قرار میدهد .



کد بالا به این صورت عمل میکند که هر خط را پردازش میکندو هش ان را میگیرد و تعداد صفر های اخر ان را شمارش میکند و در لیست ZeroCount\_list قرار میدهد هر موقع بخواهیم تعداد حانه های مورد نیاز را بدانیم از ان لیست max میگیریم .

در دیتاست اول max برابر 5 است یعنی به دو به توان پنج خانه نیاز داریم .

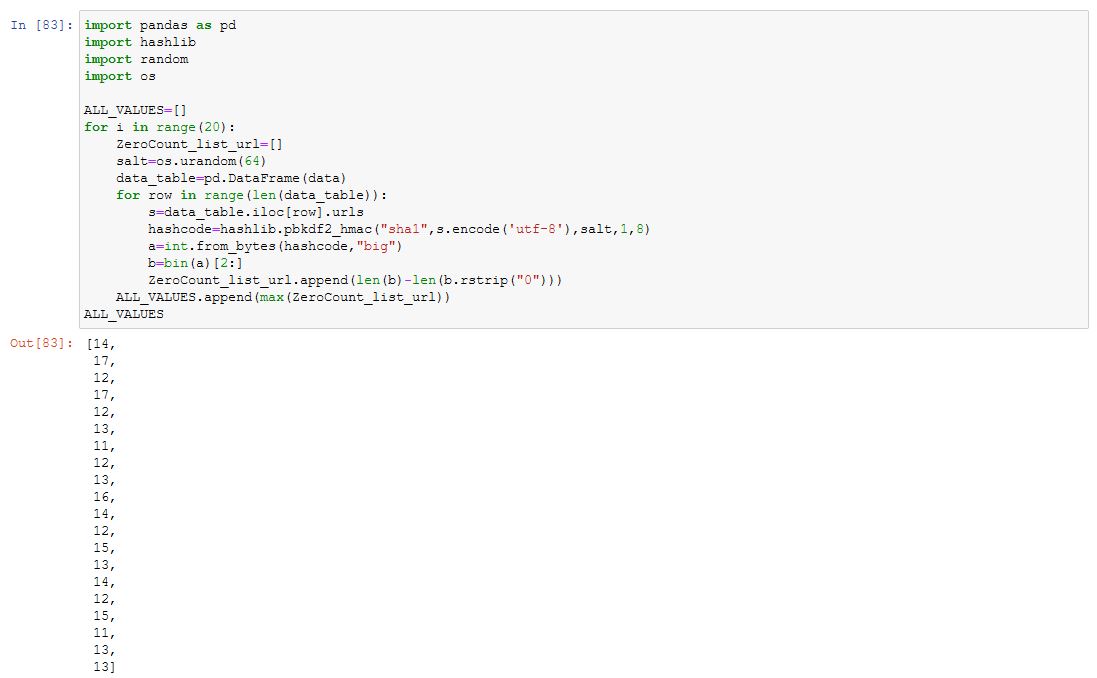
این روش را باید چند بار تکرار کنیم و از اعداد به دست امده میانگین یا میانه بگیریم .

دی دیتاست دوم :

دیتاست دوم چون csv است از پانداس برای خواندن ان استفاده میکنیم .



بعد از خواندن ان به ترتیب شروع به خواندن میکنیم و به تابع هش پاس میدهیم ان نیز ان ها را هش کرده و تعداد صفر های ان را میشمارد و در لیست قرار میدهد .

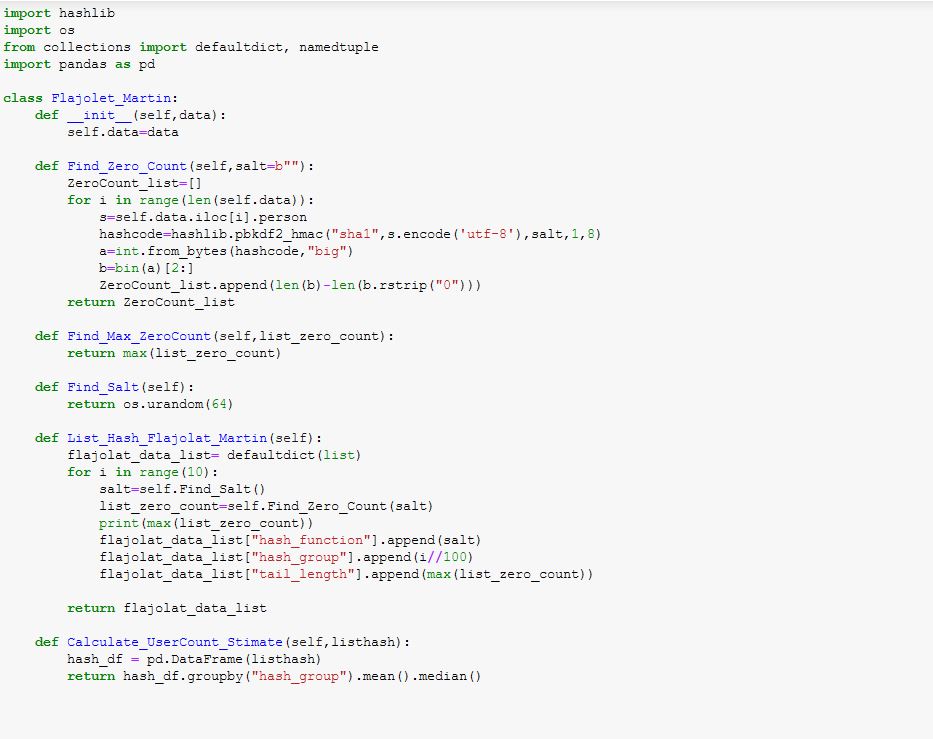


در کد بالا نمایان است که با پانداس ارایه را خوانده و تک تک تابع flajolet\_martin را روی ان اجرا میکنیم و در نهایت اگر بخواهیم تعداد r را بدست اوریم از ان ارایه max را نگه میداریم .

در دیتاست دوم این عدد 12 بدست میآید که برابر است با 2^12

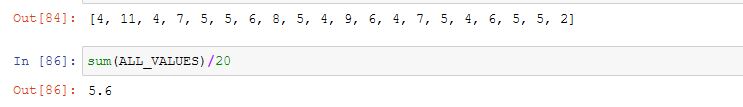
این روش را باید چند بار تکرار کنیم و از اعداد به دست امده میانگین یا میانه بگیریم .

یک کلاس کلی برای این روش نوشتم که در زیر قرار میدهم که کد نویسی تمیز در ان رعایت شده است .



من برای هر سوال 20 تا تابع هش متفاوت را اجرا کردم و خروجی هر یک را در زیر ان نشان دادم همان طور که میبینید اکثرا در یک بازه عددی خروجی میدهد ما میتوانیم با میانگین گرفتن ان مقدار بهتر ان را بدست اوریم برای مثال در اولی لیست به صورت زیر است .

[4, 11, 4, 7, 5, 5, 6, 8, 5, 4, 9, 6, 4, 7, 5, 4, 6, 5, 5, 2]



میانگین ان همان طور که میبینید 5.6 میباشد بس تعداد 2^5 تا مقدار Distinc در این ارایه وجود دارد .