مبانی فناوری بلاکچین و رمزارزها

نیم‌سال اول ۱۳۹۹-۱۴۰۰

مدرس: دکتر محمد علی مداح علی
دانشکده مهندسی برق

گزارش تمرین سری اول

نام و نام‌خانوادگی: عرفان نصرتی
شماره دانشجویی: ۹۷۱۰۲۵۵۸

مقدمه

در این تمرین از تابع hash.sha۲۵۶ به عنوان تابع hash استفاده کرده ایم که چند نکته کلی را درباره آن در اینجا آورده ایم تا از تکرار آن در هر سوال بپرسیم.اول تابع hash.sha۲۵۶ را از کتابخانه hashlib استفاده می‌کنیم. در این تابع string ها باید فرمت UTF-۸ باشد پس اولین قدم تبدیل تابع به فرمت استاندارد است که با دستور encode('UTF-۸') انجام میدهیم.در قدم بعد hexdigest() است که تابع را در مبنای hex به صورت رشته تحویل می‌دهد.

۱ سوال اول

۱.۱ قسمت اول

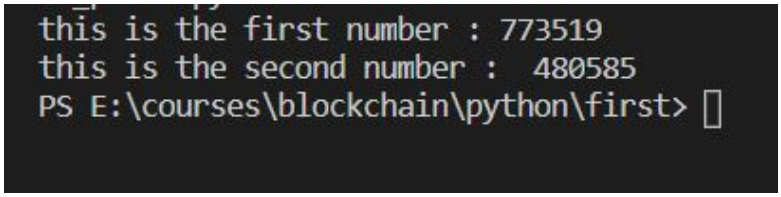
در سوال اول ما باید یک عدد پیدا کنیم که Hash آن در مبنای 2^20 با Hash شماره دانشجوییمان برابر باشد. برای اینکاز اول Hash شماره دانشجویی خود را در مبنای خواسته شده حساب می‌کنیم. سپس یک حلقه را از صفر تا2^20 می‌پیماییم تا عددی با Hash برابر شماره دانشجوییمان پیدا کنیم و عدد خواسته شده را به همراه Hash آن عدد(برای تایید یکی بودن Hash) در خروجی چاپ می کنیم که این عدد ۸۰۵۱۱۲ است.

نمونه خروجی

```
my id is :  97102558
hash of my student number is : cb87
the random number is : 805112
hash of the random number is : cb87
```

۲.۱ قسمت دوم

در قسمت دوم خواسته شده ۲ عدد با هش یکسان پیدا کنیم. طبق اصل لانه کبوتری برای داشتن دو نفر با جشن تولد یکسان باید حداقل ۳۶۷ نفر با احتساب سال کبیسه باید وحود داشته باشند تا به طور قطعی مطمئن شویم که دو نفر با تولد یکسان وجود دارند. اما مساله ی birthday paradox می‌گویید برای داشتن دو نفر با تولد یکسان با احتمال ۵.۰ تنها ۲۳ نفر و برای داشتن دو نفر با تولد یکسان با احتمال ۹.۹۹ ۷۰ نفر لازم داریم . که این به خاطر این است که احتمال انتخاب ۲ از ۲۳ نفر برابر ۲۵۳ حالت مختلف است و تعداد روز ها ۳۶۷ روز است . حال با فرمول Square approximation تعداد مراحل لازم برای پیدا کردن ۲ عدد با Hash یکسان با احتمال ۹.۰ درصد را حساب می‌کنیم (متغییر n در کد) و سپس یک حلقه به تعداد بالا میزنیم و هر بار عدد رندومی بین ۰ تا 2^20 پیدا می‌کنیم و چک میکنیم اگر Hash اعداد قبلی که حساب کرده بودیم با آن یکی بود کار را تمام می‌کنیم و دو عدد رندم همراه Hash های آن ها را چاپ می کنیم.دوباره حلقه را تکرار و عدد رندوم جدیدی پیدا می کنیم.
نمونه خروجی

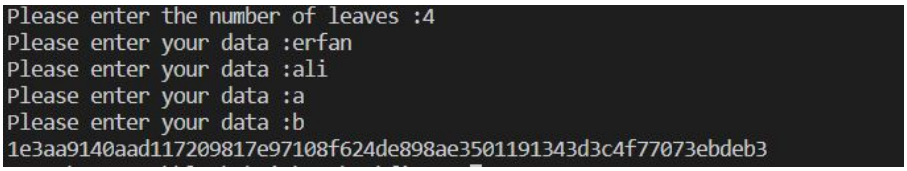


۳.۱ قسمت سوم

در قسمت سوم از ما خواسته شده که روشی برای چک کردن احراز هویت بین فرستنده و گیرنده پیام که یک پیام مشترک از قبل دارند بیان کنیم. برای این کار کافیست فرستنده پیام مشترک را در زیر پیام اصلی اضافه کند و مجموعه هر دوی آنها را Hash بگیرد و این Hash را همراه پیام برای گیرنده بفرست. گیرنده پیام دریافتی از فرستنده را که حاوی دو بخش پیام و Hash پیام+پیام مشترک است را دریافت می‌کند. برای اطمینان از اینکه پیام را حتما شخص مورد نظر فرستاده است کافیست به پیام متنی پیام مشترک را را اضافه کند و با Hash ارسالی از گیرنده مقایسه کند اگر این دو با یکدیگر برابر بودند می‌تواند مطمئن باشد تا زمانی که فرد دیگری به جز فرستنده مورد نظر پیام مشترک را نداشته باشد پیام را کس دیگری نفرستاده است یا تغییر نکرده است. اما اگر Hash حاصل با Hash قرستاده شده برابر نبود پیام $the\ message\ has\ beenchanged$ را ارسال کند. خلاصه مد به این صورت است که دو تابع sender و receiver تعریف می‌کنیم و در هر یک از این دو مراحلی که در بالا ذکر شد را انجام می‌دهیم.
نمونه خروجی

۲ سوال دوم

برای خل این سوال ابتدا نزدیکترین توان ۲ را به عدد مورد نظر پیدا می‌کنیم و اگر تعداد برگ ها کمتر از تعداد نزدیک ترین توان دو بود درخت را با الگوریتم داده شده پر می‌کنیم. سپس Hash دیتا ها را حساب میکنیم و در مرحله بعد Hash هر خانه را با خانه بعد آن حساب می کنیم و در خانه اول می گذاریم سپس Hash خانه های را ۰ با ۲ و ۴ با ۶ و ... را پیدا می کنیم و در خانه های ۰ و ۴ و .. قرار میدهیم در مرحله ی بعد قدم های خود را دو برابر کرده و خانه ۰ با ۴ و ۸ با ۱۲ و ... را با هم Hash گرفته و تا جایی که قدم ها به یک دوم تعداد برگ ها رسید ادامه می‌دهیم(یا به عبارت دیگر اگر این الگوریتم را به تعداد Log(n) بار انجام دهیم .) پس از اجرای کامل الگوریتم محتوای خانه صفرم لیست برابرmerkle root مورد نظر ما است و آنرا چاپ می‌کنیم.
نمونه خروجی



۳ سوال سوم

ایده برای حل این سوال ایده خود merkle root است در اینجا ما برای پیدا کردن ریشه از برگ ها به سمت بالا حرکت می‌کنیم تا ریشه را درست کنیم به این صورت که داده ها همراه Hash ها و ریشه را از ورودی دریافت کرده و در هر مرحله Hash ها را به یکدیگر میچسپانیم و دوباره از آن Hash می‌گیریم تا به ریشه برسیم سپس بررسی می کنیم که با آن برابر است یا خیر. در کد ابندا تعداد را گرفته سپس دیتایی که برای چک کردن میخواهیم. سپس دیتا را Hash گرفته سپس log(n) ورودی دیگر میگیریم Hash های طبقه های مختلفmerkle root است . و یک flag نیز داریم تا ببیتین باید دیتا را از سمت راست یا جپ با Hash طیقه بعد بجسپانیم زیرا اگر زوج باشد باید از سمت راست یچسپانیم و اگر فرد باشد از سمت چپ. در آخر نیز Hash را با Hash ورودی مقایسه کرده اگر برابر بود دیتا موجود است اگر نبود این دینا وجود ندارد.
نمونه خروجی

```
Please enter the number of leaves :4
which document do you want to check :2
Please enter your data :ali
please enter your layer 1 hash :8925f20dc372cb1275d7038da5f8e987571eeab194e31addbe409f63736b3246
please enter your layer 2 hash :62af5c3cb8da3e4f25061e829ebeea5c7513c54949115b1acc225930a90154da
please enter your root :1e3aa9140aad117209817e97108f624de898ae3501191343d3c4f77073ebdeb3
Existed
```