**Part 2:**

Estimation:

Since the unlock code is 6 digits long, and digits from 0 through 9 are accepted, then there are 10 possible options for each digit. This means that there are 10^6 possible combinations of a 6-digit number, or one million possible combinations. The unlock code is equal to just one of these combinations. However, the intruder does not know how long the key is, so they will be generating random numbers one at a time until the key is found. The change of guessing a 6-digit code with 10 valid options for each digit is equal to 1/10^6, which is equal to 0.000001. With this information, I estimate that the intruder will generate on average somewhere around 50000 digits before the access code is generated. This is because the intruder is highly likely to break the lock after entering one million digits, so I estimate that on average it takes half as many as this (Note, the intruder is not guaranteed to break the lock after one million digits, since they do not know it is a 6-digit access code and would therefore test other lengths, wasting digits and time). At one second per digit, I estimate that this takes 500000 seconds, or 5.788 days.

The test carried out:

I wrote a simple java program that uses the java Random class to generate one million random digits from 0 to 9 and write them to a file. I then used this file to find how many characters were generated before the unlock code of my lock (832001) was generated.

After completing the test:

Trial 1: 71292 digits were generated before the unlock code was generated

Trial 2: 109805 digits were generated before the unlock code was generated

Trial 3: 653201 digits were generated before the unlock code was generated

Trial 4: 97124 digits were generated before the unlock code was generated

Trial 5: 535618 digits were generated before the unlock code was generated

Trial 6: 54730 digits were generated before the unlock code was generated

Minimum: 54730 digits

Maximum: 653201 digits

Average: 253628 digits

Analysis:

This is very different from the average I estimated above. However, there is much to note in this data. The Minimum and maximum are quite spread out, as are the data points themselves. The data for trial 5 comes closest to my estimate. the reason for the discrepancy is that since the intruder does not know the code is only 6 digits, the data becomes a lot less predictable. If the intruder did know about the length of the code, then they are only going to try 6-digit combinations. Without that restriction, since the input is random, the result varies wildly between trials. Even though a max of 653201 digits was recorded, in each situation one million digits were generated in a matter of seconds. This means that due to how quickly even the maximum number of digits can be generated, the lock is extremely easy to break. However, if the intruder does need to wait one second between entering each digit, then it would take an average of 253628 seconds (2.94 days) according to this data. Even a relatively weak lock such as this one takes notable time to break when the speed of the input is limited. Additionally, it is worth noting that the lock could be made much stronger by incorporating letters as well as numbers into the code. If we added the entire English alphabet to the possible characters, the possible combinations increase from 10^6 to 36^6 and the chance to break the lock on one guess of a 6 digit input decreases to 1/36^6.