https://github.com/liupan15/cryptology

第二题

如何运行

```
make test
./test.exe #用于测试SM3的性能  需要一个test.txt文件


make find # 用于生成寻找 x bit的碰撞 x为输入的值
./find.exe
```

结果展示

1. 对于SM3的性能，排除读文件的时间，对于一个100MB的文件，所需要的时间为
   END RESULT: 76001 (0.01MS) 即为0.76s
   速度为 131.578947368 MB/s 左右

2. 对于寻找碰撞，我尝试寻找了60bit和64bit的碰撞，结果如下：
   对于60bit：

   找到的碰撞为：（前60bit为输入，16进制）

   d1c0a3ae 52536e5a 2cb9bfed ac8e14e7 be2113f7 65eb368c eda4ff37 a9bf0f5c

   7e1868ac 1affa545 ccdb5bb2 e8f14f5a bc6d738e c8c91e70 6d022e53 22fad00b

   得到的结果为:

   0697c841 7ceaea6a 1a4a900d 17e72f9c f847476c 8fd297df a76dc7c8 133192fa

   0697c841 7ceaea66 7a44acca 3b8d2423 c787f944 4f5daddb

77b31f1d 35f7916b

对于64bit:

找到的碰撞为：（前64bit为输入，16进制）

ce768f42 af4f6dd9 21acd20d 5317b006 ec682e54 6a33c4ff 7adf5d99 0e252ec6

23c83b83 a7cf110e 5c0eecd1 7d796bf6 2cd35a58 13c2943c 8bc73d4e 092919d4

得到的结果为：

1e290904 1d6bddca ef1743d3 e0a414fd 0891fb06 c7fc1e2c a7b2f990 48e3fbc9

1e290904 1d6bddca 4136e2ec c2a691b9 d3c0397e a4cbdf2d e4156980 33169ef8

算法简介

算法的实现参考
http://www.oscca.gov.cn/sca/xxgk/2010-12/17/1002389/files/302a3ada057c4a73830536d03e683110.pdf

按照其中的说明实现即可。

对于寻找碰撞，使用Floyd寻圈算法。
对于60bit的碰撞，初始化使用60bit的0作为输入，然后每次使用输出的256bit的前60个bit作为下一次的输入，然后按照Floyd寻找碰撞即可。（注意不能用256bit作为输入，然后比较输出的60bit是否相同）。

对于64bit的碰撞，同理。。。

第三题

利用BM算法可以得到：
f(x) = 1 + x + x^2 + x^4 + x^5
l = 7

运行BM.cpp可得

step 0: 1
l = 0
step 1: 1
l = 0
step 2: 1
l = 0
step 3: 1
l = 3
step 4: 1
l = 3
step 5: 1 + x^2
l = 3
step 6: 1 + x^2
l = 3
step 7: 1 + x^2
l = 3
step 8: 1 + x^2
l = 3
step 9: 1 + x^2
l = 6
step 10: 1 + x^1 + x^2 + x^3
l = 6
step 11: 1 + x^1 + x^2 + x^3
l = 6
step 12: 1 + x^1 + x^2 + x^5
l = 6
step 13: 1 + x^1 + x^2 + x^4 + x^5
l = 7
step 14: 1 + x^1 + x^2 + x^4 + x^5
l = 7