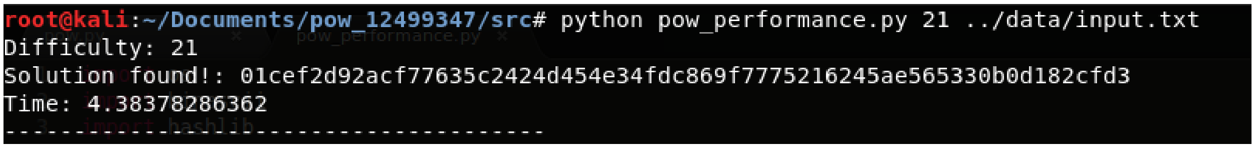
Data Security & Privacy
Project 4

Raghu Pusapati

pusaparv@mail.uc.edu

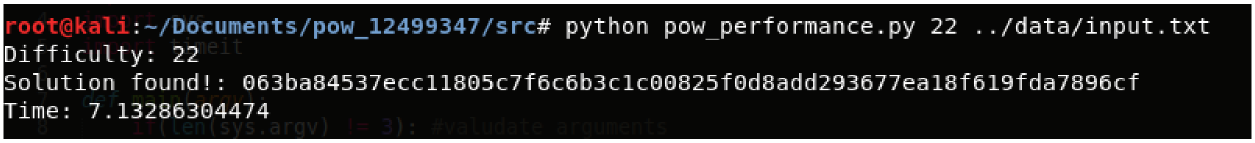
A new program, pow_performance.py has been written that computes the execution time of solution finding snippet. The following images show the outputs for difficulties from 21 through 26.

Difficulty 21:



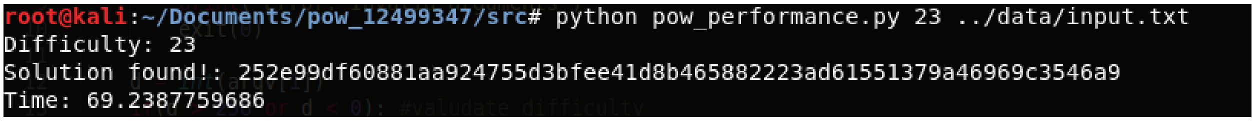
```
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 21 ../data/input.txt
Difficulty: 21
Solution found!: 01cef2d92acf77635c2424d454e34fdc869f7775216245ae565330b0d182cfd3
Time: 4.38378286362
---------------------------------------
```

Difficulty 22:



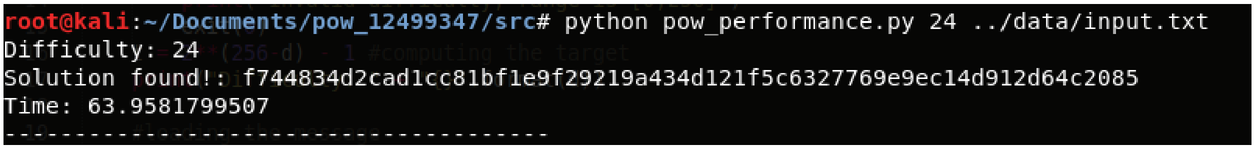
```
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 22 ../data/input.txt
Difficulty: 22
Solution found!: 063ba84537ecc11805c7f6c6b3c1c00825f0d8add293677ea18f619fda7896cf
Time: 7.13286304474
```

Difficulty 23:



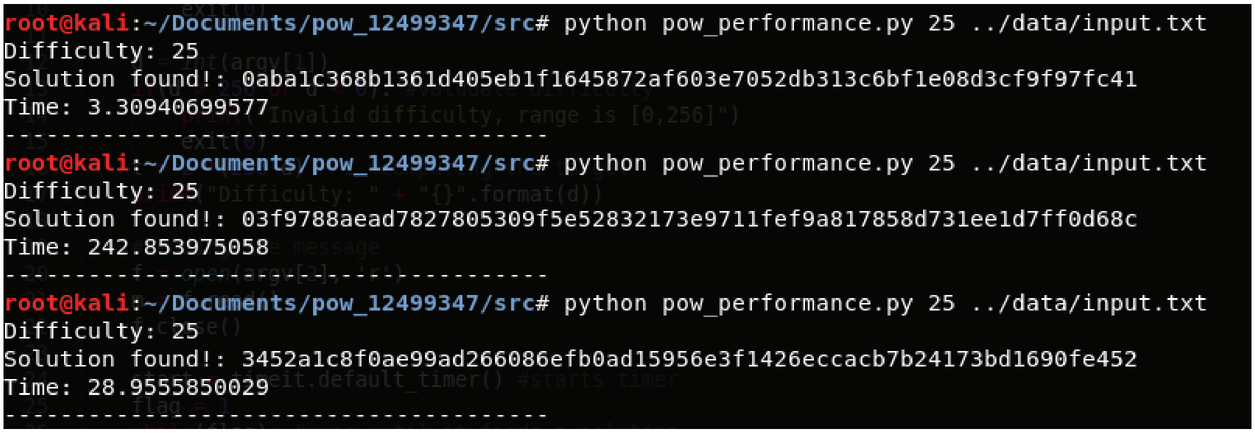
```
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 23 ../data/input.txt
Difficulty: 23
Solution found!: 252e99df60881aa924755d3bfee41d8b465882223ad61551379a46969c3546a9
Time: 69.2387759686
```

Difficulty 24:



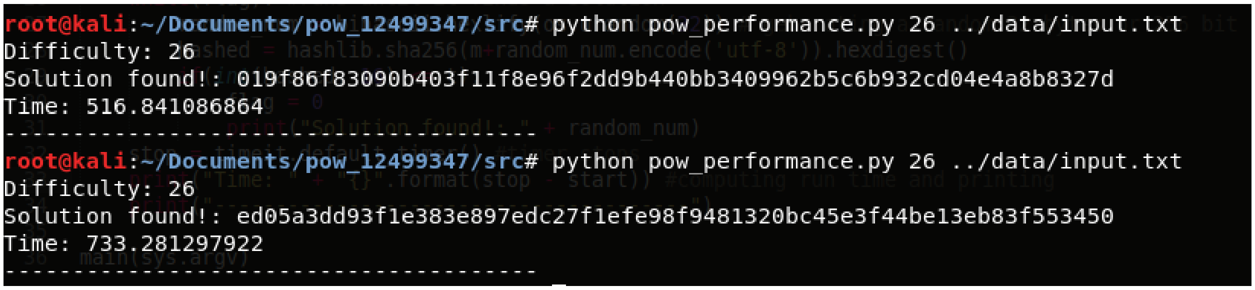
```
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 24 ../data/input.txt
Difficulty: 24
Solution found!: f744834d2cad1cc81bf1e9f29219a434d121f5c6327769e9ec14d912d64c2085
Time: 63.9581799507
---------------------------------------
```

Difficulty 25:



```
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 25 ../data/input.txt
Difficulty: 25
Solution found!: 0aba1c368b1361d405eb1f1645872af603e7052db313c6bf1e08d3cf9f97fc41
Time: 3.30940699577
---------------------------------------
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 25 ../data/input.txt
Difficulty: 25
Solution found!: 03f9788aead7827805309f5e52832173e9711fef9a817858d731ee1d7ff0d68c
Time: 242.853975058
---------------------------------------
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 25 ../data/input.txt
Difficulty: 25
Solution found!: 3452a1c8f0ae99ad266086efb0ad15956e3f1426eccacb7b24173bd1690fe452
Time: 28.9555850029
---------------------------------------
```

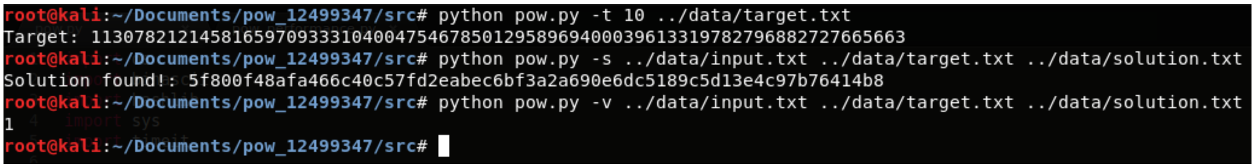
Difficulty 26:



```
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 26 ../data/input.txt
Difficulty: 26
Solution found!: 019f86f83090b403f11f8e96f2dd9b440bb3409962b5c6b932cd04e4a8b8327d
Time: 516.841086864
---------------------------------------
root@kali:~/Documents/pow_12499347/src# python pow_performance.py 26 ../data/input.txt
Difficulty: 26
Solution found!: ed05a3dd93f1e383e897edc27f1efe98f9481320bc45e3f44be13eb83f553450
Time: 733.281297922
---------------------------------------
```

We can see that time taken to find the solution generally increases with the difficulty. But sometimes one could get lucky and stumble upon the solution a lot earlier than expected like it happened for difficulty 25 as shown above. For every 1 count of increment in d, the target value nearly halves. Meaning that from 21 to 22, the target gets nearly halved, making the difficulty two times harder. So, from 21 through 26, the difficulty becomes 32 nearly times of what is it at d = 21.

The following is an example run of the PoW implementation:



```
root@kali:~/Documents/pow_12499347/src# python pow.py -t 10 ../data/target.txt
Target: 113078212145816597093331040047546785012958969400039613319782796882727665663
root@kali:~/Documents/pow_12499347/src# python pow.py -s ../data/input.txt ../data/target.txt ../data/solution.txt
Solution found!: 5f800f48afa466c40c57fd2eabec6bf3a2a690e6dc5189c5d13e4c97b76414b8
root@kali:~/Documents/pow_12499347/src# python pow.py -v ../data/input.txt ../data/target.txt ../data/solution.txt
1
root@kali:~/Documents/pow_12499347/src# █
```