Bitcoin Mining 숙제

1. 다음과 같은 인스턴스 변수와 메소드를 갖는 BloomFilter 클래스를 구현하라.

구분	속성	설명
인스턴스 변수	m	Filter의 크기 (비트 수)
	k	해시 함수의 수
	n	Filter에 저장된 항목의 수
	bf	Filter 비트맵 (BitMap 클래스를 사용)
메소드	__init__(self, m, k)	생성자 함수
	getPositions(self, item)	item을 이용하여 비트맵의 위치를 나타내는 k개의 해시 함수 값을 생성한 후, 이들의 리스트를 반환 해시 함수: sha256(item + str(1) ~ str(k)) % m
	add(self, item)	getPositions()에 의해 생성된 위치의 비트를 1로 set
	contains(self, item)	getPositions()에 의해 생성된 모드 위치의 비트가 1일 경우 True, 아니면 False를 반환
	reset(self)	비트맵의 모든 비트를 0으로 clear. n = 0
	__repr__(self)	Filter 정보 출력(m, k, 비트맵, n, 1인 비트수 등)

BitMap 클래스의 설치 및 사용 방법은 https://pypi.org/project/bitmap/ 를 참조한다.

프로그램의 실행 방법과 결과의 예는 다음과 같다.

```
if __name__ == "__main__":
    bf = BloomFilter(53, 3)
    for ch in "AEIOU":
        bf.add(ch)
    print(bf)
    for ch in "ABCDEFGHIJ":
        print(ch, bf.contains(ch))
```

```
M = 53, F = 3
BitMap = 00010010001000000100000000001110000001000010000100001000
항목의 수 = 5, 1인 비트수 = 11
A True
B False
C True      ← false positive
D False
E True
F False
G False
H False
I True
J False
```

(다음 페이지에 계속)

2. Bitcoin의 POW를 구현한 후, 메시지와 target bits를 입력받아 POW를 만족하는 nonce 값을 출력하라. nonce는 4바이트의 정수를 사용한다. POW를 만족하는 nonce 값이 존재하지 않을 수 있으므로, 현재 시각을 extra nonce로 사용하며 POW를 만족하는 nonce가 없을 때 변경한다. 즉, POW는 다음과 같다.

   SHA256(메시지 + extra nonce + nonce) < Target

○ 입력의 예:

- 메시지의 내용? 학번=123456
- Target bits? 1e00ffff

○ 출력의 예:

- Target: 0x000000ffff0000000000000000000000000000000000000000000000000000
- 메시지: 학번=123456, Extra nonce: 1680068520, nonce: 13367652
- 실행 시간: 22.305328607559204초
- Hash result: 0x0000008150ae22aafd7fdc74a5faa2b064fce6e4557ee6c35211e86452205c5e