

Работаем с адресами и масками

Представьте себе, что вас приняли младшим хакером в банду, осуществляющую DDoS-атаки.

Вам поручили сгенерировать список фальшивых IP-подсетей для спуфинга. Требования к подсетям:

- Подсеть должна выбираться случайным образом
- Диапазон — между 11.0.0.0 (0x0B000000) и 223.0.0.0 (0xDF000000)
- Диапазоны адресов должны быть реальными (не частными)
- Маска — между /8 и /24

Результат 1: создать новый класс, решающий предложенную задачу (методом наследования от подходящего класса библиотеки `ipaddress`)

Результат 2: список подсетей (для определенности — 50 штук), отсортированных сначала по маске (/8 в самом начале, /24 в самом конце), затем по адресу

- Подключить библиотеки `ipaddress` и `random`
- Создать класс `IPv4RandomNetwork`:
 - должен быть унаследован от класса `IPv4Network` библиотеки `ipaddress`
 - должен содержать метод `regular`, возвращающий `True`, если сеть обычная, и `False` — в остальных случаях (воспользуйтесь полями базового класса `IPv4Network.is_*`)
 - должен содержать метод, возвращающий число —комбинацию из IP-адреса и маски для последующей сортировки
 - при создании объекта класса должен генерировать случайную сеть со случайной длиной префикса
- Сгенерировать список сетей `IPv4`, вывести на экран в отсортированном виде
 - используйте функцию `sorted`
 - задайте функцию сортировки и передайте её в качестве параметра `key` в функцию `sorted`

```
from ipaddress import IPv4Network
import random

class IPv4RandomNetwork(IPv4Network):
    def __init__(self, ...):
        ...
        IPv4Network.__init__(self, ...)

    def key_value(self):
        return ...
```

int(IPv4Network.network_address) — получить адрес подсети в виде числа

int(IPv4Network.netmask) — получить маску подсети в виде числа

sorted(последовательность, key=функция) — возвращает отсортированную последовательность

Проверка работы:

```
$ git clone git://github.com/<username>/p4ne  
...  
$ python ./p4ne/Lab1.4/ipaddr.py
```

Должен появиться список IP-адресов с масками, отсортированный сначала по длине префикса (короткие вверху), затем по значению