**Тестовое задание**

**Постановка задачи**:

Реализовать на языке программирования Python приложение для нахождения минимальной подсети для заданного набора IP-адресов.

**Необходимо реализовать**

1. Приложение, подбирающее для заданного набора IP-адресов минимальную подсеть, содержащую данные IP-адреса.
2. Набор тестов для регрессионного тестирования и проверки corner cases (например, некорректные или несоответствующие заданной версии IP-адреса в файле).
3. Документирование кода и приложения.

**Обязательные требования**:

1. Наборы IP-адресов хранятся в текстовых файлах (отдельно для IPv4 и IPv6).
2. Программа запускается через командную строку (CLI), принимая в качестве параметров имя файла с IP-адресами и версию IP.
3. Используемые функции и алгоритмы снабжены информативными комментариями.
4. Тесты оформлены в отдельном файле.
5. Документация в виде README.md файла содержит оценки временнОй эффективности используемого алгоритма и описание способов запуска программы.
6. Соответствие кода PEP8 и использование аннотаций типов.
7. Использование внешних зависимостей (модулей, не входящих в стандартную библиотеку) свести к минимуму.

**Допущения**:

1. Можно ограничиться поддержкой IPv4.
2. Тестовое покрытие оценивать не обязательно, но желательно.

**Требования к инструментам**:

1. Язык программирования – Python (версия не старше 3.6).
2. Исходный код должен быть опубликован на Github/Gitlab/Bitbucket.
3. (Будет плюсом) создание установочного пакета с setuptools.

**Срок выполнения**: неделя.

**Примеры:**

Пример входных данных IPv4:

192.168.1.2

192.168.1.3

192.168.1.5

Ожидаемый результат:

Result net: 192.168.1.0/29

Пример входных данных IPv6:

ffe0::1:0:0:0

ffe0::2:0:0:0

ffe0::4:0:0:0

ffe0::8:0:0:0

ffe0::10:0:0:0

ffe0::20:0:0:0

ffe0::40:0:0:0

ffe0::80:0:0:0

Ожидаемый результат:

Result net: ffe0::/72