Категории:

Безопасность (5)

Инструменты (1)

C# (7)

Веб-разработка (15)

Информатика (158)

ЕГЭ (133)

ОГЭ (25)

Математика (10)

Физика (22)

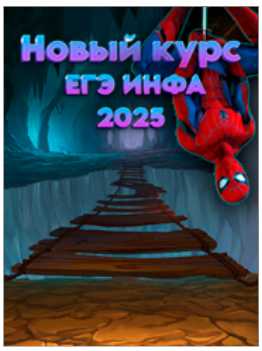
Алгоритмы (12)

Новости (9)

Задачи (10)

Разное (5)

Life (3)



Свежие комментарии:

Илья:
СПС

01-10-2024 Читать статью

ЕГЭ по информатике 2025 - Задание 13 (Работа с сетью)

В этой статье мы разберём **13 задание** из **ЕГЭ по информатике 2025** на ip адреса.

Тематика тринадцатого задания из **ЕГЭ по информатике 2025** затрагивает организацию компьютерных сетей, адресацию, протоколы передачи данных.

Перейдём непосредственно к решению типовых задач.

Задача (Классика)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и маске сети.
Сеть задана IP-адресом 192.168.32.160 и маской сети 255.255.255.240. Сколько в этой сети IP-адресов, для которых сумма единиц в двоичной записи IP-адреса чётна?
В ответе укажите только число.

Решение:

В начале задачи обычно даётся краткая теория, которая не меняется от задачи к задаче. Нас интересуют последние два абзаца.

Здесь речь идёт о локальной сети, например, которая, возможно, есть у вас в школе.

Нам дали **IP-адрес** в какой-то локальной сети и маску. По этим данным мы можем понять, какие ещё IP-адреса находятся в этой сети.

Для этого выпишем IP-адрес и под ним маску.

```
192. 168.  32.  160
255. 255. 255. 240
```

Каждое число, отделённое точкой, **весит 1 байт** (8 бит). Максимальное числовое значение может быть **255**. Представим все числа в двоичном виде. Это можно сделать с помощью стандартного калькулятора Windows в режиме "программиста" или в Python с помощью функции **format()** (см. Задание 5).

```
11000000. 10101000. 00100000. 10100000
11111111. 11111111. 11111111. 11110000
```

Если число в двоичной форме имеет меньше, чем 8 разрядов, то нужно дополнить его нулями слева до 8 разрядов. Например, число 5 = 101_2 будет представлено,

как 5 = 00000101_2.

Отметим, что в маске в начале идут единицы, и с какого-то момента начинаются нули. Т.е. **в маске не могут чередоваться нули и единцы**, есть чёткая граница между ними.

Маска показывает, какая часть IP-адреса будет у всех IP-адресов в этой сети. Эта часть находится над единицами маски. То, что находится над нулями маски, может меняться. Так как раз и формируется всё многообразие IP-адресов в этой сети.

Если мы переберём все комбинации в той части IP-адреса, которая отмечена зелёным цветом (находится на нулями маски), мы получим все IP-адреса в этой сети.

Чтобы ответить на вопрос задачи, подсчитаем, сколько единиц уже точно будет у каждого IP адреса (считаем единицы в красной области IP-адреса). Получается 8 единиц. Теперь решим данную задачу, как 8 задание. В четырёх ячейках, которые отмечены зелёным цветом, крутятся нули и единицы, нужно найти количество комбинаций, которые подходят под условие задачи.

```
k=0
for x1 in '01':
    for x2 in '01':
        for x3 in '01':
            for x4 in '01':
                s = x1 + x2 + x3 + x4
                if (8+s.count('1'))%2==0:
                    k=k+1
print(k)
```

Четыре нуля в маске, следовательно, крутим четыре цикла. Мы ищем те комбинации, чтобы суммарное количество единиц было чётно. **Получается 8 комбинаций**. Это и будет ответ.

Первая комбинация, когда в этих битах, которые изменяются и отмечены зелёным цветом, все нули, соответствует **адресу сети**. Последняя комбинация, когда в этих битах все единицы, соответствует **широковещательному адресу**. Если перевести обратно в десятичные числа эти два адреса, получится 192.168.32.160 и 192.168.32.175.

Так же **адрес сети** можно получить, применив поразрядное умножение (конъюнкцию) между двоичным представлением любого IP-адреса сети и двоичным представлением маски. Макса просто занулит те биты IP-адреса, где сама имеет нули.

На практике эти два адреса (адрес сети и широковещательный адрес) не раздаются устройствам в сети. По-другому физические устройства называют ещё **узлами сети**. Эти два IP-адреса используются в технических целях.

Данный способ имеет ограничение. Максимум можно сделать 20 вложенных циклов. Если количество нулей в маске превышает это значение, нужно пользоваться другим способом.

Решение через модуль ipaddress

```
from ipaddress import *

net = ip_network('192.168.32.160/255.255.255.240', 0)

for ip in net:
    print(ip)
```

Здесь мы подключили модуль **ipaddress** для работы с IP-адресами. Создали сеть с помощью функции **ip_network**. В начале нужно писать **адрес сети**. Через слеш пишем маску. Так же написали ещё один параметр, который равен нулю. Это сделали потому, что мы не знаем изначально, является ли наш IP-адрес именно **адресом сети**. Чтобы программа автоматически нашла адрес сети и сформировала по нему сеть, мы поставили ещё один параметр 0.

Программа выведет следующие ip-адреса:

192.168.32.160
192.168.32.161
192.168.32.162
192.168.32.163
192.168.32.164
192.168.32.165
192.168.32.166
192.168.32.167
192.168.32.168
192.168.32.169
192.168.32.170
192.168.32.171
192.168.32.172
192.168.32.173
192.168.32.174
192.168.32.175

Первый адрес из этого списка - это адрес сети, последний - это широковещательный. Если мы хотим от них избавиться, то нужно добавить **.hosts()**: ip_network('192.168.32.160/255.255.255.240', 0).hosts()

В данной задаче про эти два адреса ничего не сказано.

Остаётся немного докрутить, чтобы решить задачу.

```
from ipaddress import *

net =  ip_network('192.168.32.160/255.255.255.240', 0)

for ip in net:
    if format(ip, 'b').count('1')%2==0:
        print(ip)
```

Получается так же 8 IP-адресов, которые удовлетворяют условию задачи. Функция **format()** принимает именно ip-адрес и правильно преобразует его в двоичный вид в виде строки.

192.168.32.160
192.168.32.163
192.168.32.165
192.168.32.166
192.168.32.169
192.168.32.170
192.168.32.172
192.168.32.175

Можно не печатать адреса, а поставить счётчик.

```
from ipaddress import *

k=0

net = ip_network('192.168.32.160/255.255.255.240', 0)

for ip in net:
    if format(ip, 'b').count('1')%2==0:
```

```
        k += 1

print(k)
```

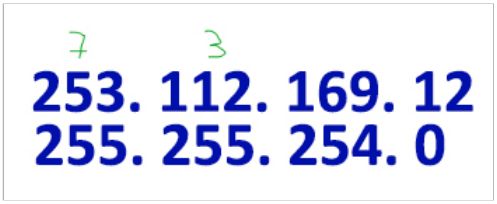
Ответ: 8

Задача (Два левых, два правых байта)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и маске сети.
Сеть задана IP-адресом 253.112.169.12 и маской сети 255.255.254.0. Сколько в этой сети IP-адресов, для которых в двоичной записи IP-адреса суммарное количество единиц в правых двух байтах не менее суммарного количества единиц в левых двух байтах.

Решение:

Выпишем IP-адрес и под ним маску. Первый байт слева будет у всех IP-адресов в этой сети одинаковый, т.к. в маске стоит число 255, это все единицы в двоичной системе 255 = 11111111_2. А что находится над единицами маски - не меняется. Выпишем сразу количество единиц для первого байта IP-адреса. Их будет 7. Тоже самое сделаем и со вторым байтом. Там будет 3 единицы.



Получается, что в левых двух байтах количество единиц равно 10 (7+3).

В третьем же байте появляются нули в маске. Рассмотрим третий и четвёртый байты более подробно в двоичной системе.

Мы видим, что в правых двух байтах уже точно будут 3 единицы во всех IP-адресах сети. Прокрутим зелёную область (над нулями маски), как в 8 задании.

```
k=0
for x1 in '01':
    for x2 in '01':
        for x3 in '01':
            for x4 in '01':
                for x5 in '01':
                    for x6 in '01':
                        for x7 in '01':
                            for x8 in '01':
                                for x9 in '01':
                                    s = x1+x2+x3+x4+x5+x6+x7+x8+x9
                                    if 10 <= 3 + s.count('1'):
                                        k += 1

print(k)
```

Ответ получается **46**.

Решим задачу с помощью модуля ipaddress.

```
from ipaddress import *

k=0

net = ip_network('253.112.169.12/255.255.254.0', 0)

for ip in net:
    s = format(ip, 'b')
    if s[:16].count('1') <= s[16:].count('1'):
        k += 1

print(k)
```

Здесь мы перебираем все IP данной сети в виде строки **s**. Рассматриваем левые два байта s[:16] (первые 16 символов) и правые два байта s[16:] (последние 16 символов). Если условие задачи выполняется, то подсчитываем такой IP.

Ответ: 46

Задача (Минимальный байт маски)

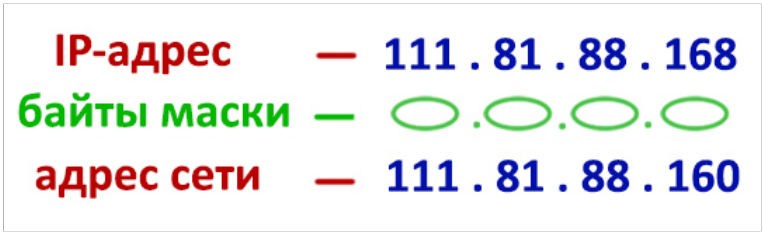
В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая - к адресу самого узла в этой сети. Обычно маска записывается по теме же правилам, что и IP-адрес - в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда - нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 111.81.88.168 адрес сети равен 111.81.88.160.

Найдите наименьшее значение последнего байта маски. Ответ запишите в виде десятичного числа.

Выпишем **IP-адрес**, а под ним **адрес сети**, пропустив свободную строчку. В свободной строчке мы должны записать **байты маски**.

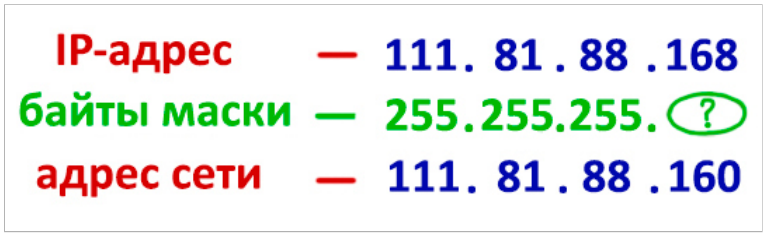


Рассмотрим левый столбик. В **IP-адресе** и в **адресе сети** одинаковое число **111**. Значит, первый слева байт маски равен числу **255**.

Если записать числа в двоичной системе в виде 8 разрядов (1 байта), то логическое умножение двоичных разрядов байта **IP-адреса** и байта **маски** должно давать байт **адреса сети**

Почему нельзя поставить в байт маски число **239** (11101111_2) ? Или число **111** (01101111_2) ? Дело в том, что в маске в начале идут единицы, а потом с какого-то момента нули. Единицы и нули в маске не чередуются!

Теперь мы понимаем, что три левых байта маски могут принимать значение только **255** (В двоичном представлении все единицы 11111111_2), из-за того, что совпадают числа **IP-адреса** и **адреса сети** в трёх левых байтах.



Значение последнего байта маски нужно проанализировать и сделать его как можно меньше, исходя из условия задачи.

Записываем последний байт **IP-адреса** и под ним, пропустив свободную строчку для байта маски, байт **адреса сети** в двоичной системе.

Т.к. нужно найти наименьшее значение для байта маски, начинаем справа забивать нулями, пока выполняется поразрядная конъюнкция. Как только невозможно будет поставить ноль, придётся ставить единицу, и дальше пойдут влево только единицы.

В шестой разряд справа уже нельзя поставить 0, потому что 1 * 0 будет 0, а должна быть 1!

Примечание: Мы забили нулями по максимуму байт маски, но так же было бы корректно байт маски представить в таком виде 11110000_2, однако такое представление **не делает** байт маски **минимальным** в числовом значении.

Переводим в десятичную систему получившийся байт маски 11100000_2.

$$11100000_2 = 224$$

В задаче сказано, что нам дали именно **IP-адрес узла**. Это значит, что при поиске байта маски, этот IP-адрес не должен совпадать с адресом сети и с широковещательным адресом. Реальным устройствам (узлам) не дают эти два технических адреса. В нашем случае всё в порядке, но это нужно иметь в виду.

Решим задачу с помощью модуля ipaddress.

```
from ipaddress import *

ip = ip_address('111.81.88.168')

for mask in range(31):
```

```
    net = ip_network(f'{ip}/{mask}', 0)
    if net[0] < ip < net[-1]:
        print(net, net.netmask)
```

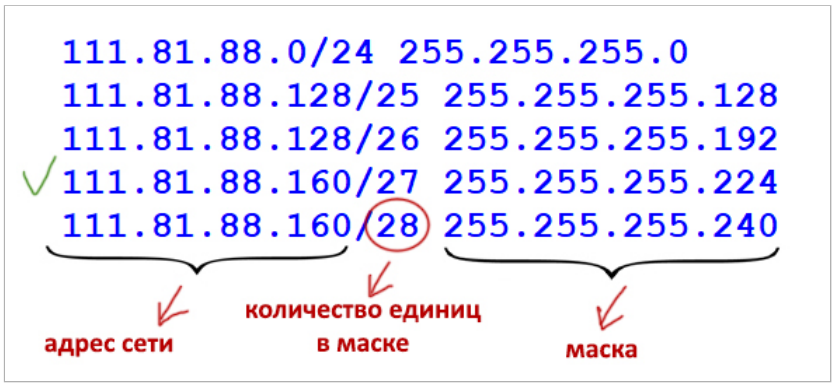
В начале **ip** перевели в специальный формат с помощью функции **ip_address()**. Затем прокручиваем маску в цикле. В данном случае это количество единиц в маске. *Можно указывать маску не в виде четырёх чисел, а через количество единиц*. Максимум в маске может быть 32 единицы. Но мы крутим до 30 включительно, т.к. 31 единица и 32 единицы - это неприменимые маски, их можно не рассматривать.

Для каждой маски формируем сеть **net**. Символ **f** означает, что внутри строки можно писать название переменных в фигурных скобках.

Мы проверяем исходный ip, чтобы в этой сети он не совпадал с адресом сети (он идёт самый первый net[0]) и с широковещательным адресом (он идёт последним net[-1]). Нужно, чтобы он находился между ними.

Если всё сходится, печатаем сеть и адрес сети.

Программа распечатает разные варианты сетей. Рассмотрим конец таблицы. Нужно найти такую сеть, чтобы адрес сети совпадал с нашим адресом.



Это последняя и предпоследняя сеть. Но выбираем предпоследнюю сеть, т.к. в ней последний байт маски будет с наименьшим числовым значением.

Ответ: 224

Задача (Максимальное количество единиц в маске)

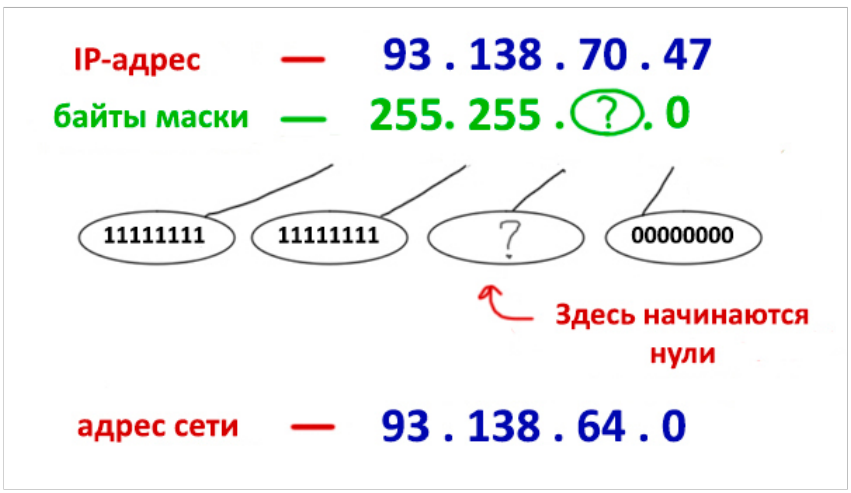
В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места – нули. Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

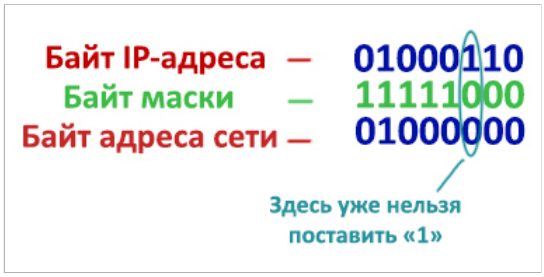
Для узла с IP-адресом 93.138.70.47 адрес сети равен 93.138.64.0. Каково наибольшее возможное общее количество единиц во всех четырёх байтах маски? Ответ запишите в виде десятичного числа.

Решение:

Напишем общую ситуацию для **IP-адреса** и **адреса сети**.



Переведём числа **70** и **64** в двоичную систему, чтобы узнать второй справа байт маски.



Начинаем забивать единицы слева в байте маске. В 5 разрядах слева это можно сделать, но в шестом слева разряде должны поставить 0. Если поставить единицу получится 1*1=1, а должен получится ноль в разряде **адреса сети**.

Нули так же распространяются на последний байт маски.

Нужно убедиться, что IP-адрес узла при данной маске не совпадает с адресом сети и с широковещательным адресом в этой сети. У нас это выполняется.

Во втором справа байте маски получилось, что наибольшее количество единиц равно 5. Тогда ответ будет 8 + 8 + 5 = 21 единица во всех 4 байтах маски.

Решение через модуль ipaddress

```
from ipaddress import *

ip = ip_address('93.138.70.47')

for mask in range(31):
    net = ip_network(f'{ip}/{mask}', 0)
    if net[0] < ip < net[-1]:
        print(net)
```

```
93.138.64.0/18
93.138.64.0/19
93.138.64.0/20
93.138.64.0/21
93.138.68.0/22
93.138.70.0/23
93.138.70.0/24
93.138.70.0/25
93.138.70.0/26
93.138.70.32/27
```

Ищем наш адрес сети, где максимальное количество единиц в маске.

Ответ: 21

Задача (Количество адресов компьютеров)

В терминологии сетей TCP/IP маской подсети называется 32-разрядное двоичное число, определяющее, какие именно разряды IP-адреса компьютера являются общими для всей подсети – в этих разрядах маски стоит 1. Обычно маски записываются в виде четверки десятичных чисел – по тем же правилам, что и IP-адреса. Для некоторой подсети используется маска 255.255.248.0. Сколько различных адресов компьютеров допускает эта маска?

Примечание. На практике для адресации компьютеров не используются два адреса: адрес сети и широковещательный адрес.

Решение:

Здесь нам дана только **маска** и у этой задачи совсем другой вопрос. Ключевой фразой здесь является: **"адресов компьютеров"**.

Для начала нужно узнать, сколько нулей в маске (4 байтах).

Последний (самый правый байт полностью занулён), значит, 8 нулей уже есть. Нули начинаются во втором справа байте, ведь первые два байта маски имеют значение 255, что в двоичной системе обозначает 8 единиц (11111111_2)

Переведём число **248** в двоичную систему.

Число **248** в в двоичной системе будет **11111000_2**.

Итого, во всей маске у нас получается 8 + 3 = 11 нулей!

Именно нули в маске показывают количество адресов компьютеров! Применяем формулу:

$$N = 2^{11} = 2048$$

Или можем составить программу, как в 8 Задании.

```
k=0
for x1 in '01':
    for x2 in '01':
        for x3 in '01':
            for x4 in '01':
                for x5 in '01':
                    for x6 in '01':
                        for x7 in '01':
                            for x8 in '01':
                                for x9 in '01':
                                    for x10 in '01':
                                        for x11 in '01':
                                            s = x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11
                                            k += 1

print(k)
```

В примечании сказано, что не используются два адреса из этого набора, значит в ответе запишем 2048 - 2 = **2046**.

Решение через модуль ipaddress

```
from ipaddress import *
net = ip_network('0.0.0.0/255.255.248.0')
print(net.num_addresses - 2)
```

Здесь мы взяли произвольный адрес сети "0.0.0.0" и применили нашу маску. Свойство **num_addresses** покажет количество возможных адресов в этой сети. Необходимо так же вычистить два неиспользуемых адреса.

Ответ: 2046

Задача (Количество масок)

В терминологии сетей TCP/IP маска сети – это двоичное число, меньшее 2^{32}; в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места нули. Маска определяет, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырёх байт, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.
Для узла с IP-адресом 175.122.80.13 адрес подсети равен 175.122.80.0. Сколько существует различных возможных значений маски, если известно, что в этой сети не менее 60 узлов? Ответ запишите в виде десятичного числа.

Решение:

Сколько нулей должно быть точно в маске? Применяем формулу:

$$2^6-2 = 64 - 2 = 62 >= 60$$

Здесь мы отнимаем двойку, т.к. узлам (устройствам) не даются два технических адреса: адрес сети и широковещательный адрес.

Т.е. получается 6 нулей минимум. Распишем последние два байта для IP адреса узла и сети:

$$80.13 = 01010000\ .\ 00001101$$

$$80.0\ \ = 01010000\ .\ 00000000$$

Последние 6 нулей мы не имеем права трогать, т.к. они отвечают за количество узлов. Остальные можно перебирать, но важно помнить про правило: в маске в начале идут единицы, а с какого-то момента нули.

```
80.13 = 01010000.00001101
                xxxx.xx000000
80.0  = 01010000.00000000
```

Получается вместо "x" можем написать следующие комбинации:

```
0000.00
1000.00
1100.00
1110.00
1111.00
1111.10
1111.11
```

Получается **7** масок.

Решение через модуль ipaddress

```
from ipaddress import *

ip = ip_address('175.122.80.13')

for mask in range(31):
    net = ip_network(f'{ip}/{mask}', 0)
    if net[0] < ip < net[-1]:
        print(net)
```

```
175.122.64.0/18
175.122.64.0/19
175.122.80.0/20  ✓
175.122.80.0/21  ✓
175.122.80.0/22  ✓
175.122.80.0/23  ✓
175.122.80.0/24  ✓
175.122.80.0/25  ✓
175.122.80.0/26  ✓
175.122.80.0/27
175.122.80.0/28
175.122.80.8/29
175.122.80.12/30
```

Ищем те сети, где есть наш адрес сети. Но миниму может быть 6 нулей, т.е. максимум 32 - 6 = 26 единиц.

Можно воспользоваться свойством **num_addresses**.

```
from ipaddress import *

ip = ip_address('175.122.80.13')

for mask in range(31):
    net = ip_network(f'{ip}/{mask}', 0)
```

```
    if net[0] < ip < net[-1]:
        if net.num_addresses - 2 >= 60:
            print(net)
```

Ответ: 7

Задача (Порядковый номер компьютера)

Маской подсети называется 32-разрядное двоичное число, которое определяет, какая часть IP-адреса компьютера относится к адресу сети, а какая часть IP-адреса определяет адрес компьютера в подсети. В маске подсети старшие биты, отведенные в IP-адресе компьютера для адреса сети, имеют значение 1; младшие биты, отведенные в IP-адресе компьютера для адреса компьютера в подсети, имеют значение 0.
Если маска подсети 255.255.255.224 и IP-адрес компьютера в сети 162.198.0.157, то порядковый номер компьютера в сети равен_____

Решение:

В этой задаче ключевой фразой является: **"порядковый номер компьютера"**. Нужно знать, как решать данную тренировочную задачу из **ЕГЭ по информатике**.

Первые 3 слева байты маски равны 255 (11111111_2), значит, они не участвуют в решении этой задачи.

Мы фокусируем внимание на том байте IP-адреса, под которым байт маски имеет не все единицы в своих разрядах.

Переведём числа **224** и **157** в двоичную систему.

Число **224** в двоичной системе равно $\mathbf{11100000_2}$.

Число **157** в двоичной системе равно $\mathbf{10011101_2}$.

Запишем друг под другом данные числа в двоичной системе

Выписываем ту часть **IP-адреса**, которая находится над нулями.

Нужно перевести это двоичное число $\mathbf{11101_2}$ в десятичную систему, это и будет ответ. Получается число **29**

Примечание:

Предположим IP адрес будет 162.198.157.10, а маска подсети 255.255.224.0, тогда запишем байты IP-адреса, а под ними байты маски:

100**11101 00001010**
11100000 00000000

То берём всё равно ту часть ip-адреса, которая находится над нулями! Не ограничиваемся 8-ю разрядами!

$1110100001010_2 = 7434$

Данную задачу лучше решать вручную без использования **ipaddress**.

Ответ: 29

Задача (Параметр А в маске)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и маске сети.

Сеть задана IP-адресом 255.211.33.160 и маской сети 255.255.A.0, где A - некоторое допустимое для записи маски число. Определите минимальное значение A, для которого для всех IP-адресов этой сети в двоичной записи IP-адреса суммарное количество единиц в левых двух байтах не менее суммарного количества единиц в правых двух байтах.

Решение:

Решим данную задачу сразу с помощью модуля **ipaddress**.

```
from ipaddress import *

for i in range(9):

    A_bin = i * '1'

    while len(A_bin)<8:
        A_bin = A_bin + '0'

    A = int(A_bin, 2)

    net = ip_network(f'255.211.33.160/255.255.{A}.0', 0)
    flag = 1
    for ip in net:
        s = format(ip, 'b')
        if s[:16].count('1') < s[16:].count('1'):
            flag=0
            break
    if flag==1:
        print(A)
```

В начале формируем байт маски в двоичной системе **A_bin** в виде строки. Переводим это число в десятичное число **A**. Формируем сеть **net** по заданным параметрам. Переменная **flag** показывает все ли IP-адреса в этой сети подходят под условие задачи. Если хотя бы один адрес противоречит условию задачи, то **flag** будет равен нулю, иначе **flag** останется единицей.

Если все адреса подошли в данной сети под условие задачи, то печатаем значение A, при котором это произошло.

Ответ: 240

Так же смотрите в видеоролике решение похожей задачи вручную.

Задача (Параметр A в IP-адресе)

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и маске сети.

Сеть задана IP-адресом 32.0.A.5, где A - некоторое допустимое для записи IP-адреса число, и маской сети 255.255.240.0. Определите минимальное значение A, для которого для всех IP-адресов этой сети в двоичной записи IP-адреса суммарное количество единиц в левых двух байтах не более суммарного количества единиц в правых двух байтах.

Решение:

```
from ipaddress import *

for A in range(256):

    net = ip_network(f'32.0.{A}.5/255.255.240.0', 0)
    flag = 1
    for ip in net:
        s = format(ip, 'b')
        if s[:16].count('1') > s[16:].count('1'):
            flag=0
            break
    if flag==1:
        print(A)
        break
```

Здесь уже нужно перебирать байт IP-адреса, а не маски, значит, можно перебрать числа от 0 до 255.

Далее действуем, как в прошлой задаче.

Ответ: 16

Как решать вручную похожую задачу, смотрите в видеоролике ниже.

Пусть Вам повезёт при решении **13 задания** из **ЕГЭ по информатике 2025**.

Поддержать сайт:

Похожая статья:

ЕГЭ по информатике 2025 - Задание 25 (Делимость чисел)

Всем привет! Добрались мы до 25 задания из ЕГЭ по информатике 2025. Ра...

Категория: **Информатика** Подкатегория: **ЕГЭ**

Дата: **02-08-2024 в 13:40:53** 0

Комментарии:

"Если число в двоичной форме имеет меньше, чем 8 разрядов, то нужно дополнить его нулями справа до 8 разрядов. Например, число 5 = 1012 будет представлено, как 5 = 000001012." это немного слева, а не справа

9-2024 в 15:07:48

Исправил!

жский Александр 05-09-2024 в 18:07:15

Оставить коментарий:

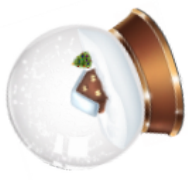
Имя

Текст

Напишите email, чтобы получать сообщения о новых комментариях (необязательно):

Email

Задача против робота. Расположите картинки горизонтально:



Отправить

Нажимая кнопку Отправить, Вы соглашаетесь с политикой конфиденциальности сайта.