

# Открытая олимпиада школьников (информатика) 2024-25

## Решения заданий заключительного этапа для 10 и 11 класса (типовой вариант)

### 1. Кодирование информации. Системы счисления (2 балла)

#### [100 первых цифр]

Петя составляет ряд из периодических дробей, записанных в шестнадцатеричной системе счисления. Первый член ряда равен  $0,(\text{AB})_{16}$ . Каждый следующий член ряда ровно в 4 раза меньше предыдущего. Известно, что у некоторого члена ряда сумма первых 100 цифр после запятой равна  $441_{10}$ . Найдите и укажите в ответ номер этого члена ряда. Если такого члена ряда не существует, запишите в ответ NULL.

**Ответ: 117**

**Решение:**

Обратим внимание, что деление числа на 4 равносильно сдвигу запятой на один разряд влево в записи числа в четверичной системе счисления. Также вспомним, что мы можем перевести число из шестнадцатеричной системы счисления в четверичную заменив независимо каждый шестнадцатеричный разряд на два четверичных.

Тогда первый член ряда  $0,(\text{AB})_{16} = 0,(2223)_4$ .

$0,(2223)_4 / 4_{10} = 0,(2223)_4 / 10_4 = 0,0(2223)_4 = 0,2(\text{AE})_{16}$  – это второй член ряда.

Повторим операцию деления несколько раз, зафиксировав закономерность в записи получающихся членах ряда.

$0,0(2223)_4 / 4_{10} = 0,0(2223)_4 / 10_4 = 0,00(2223)_4 = 0,0(\text{AB})_{16}$  – третий член ряда.

$0,00(2223)_4 / 4_{10} = 0,00(2223)_4 / 10_4 = 0,000(2223)_4 = 0,02(\text{AE})_{16}$  – четвертый член ряда.

Легко заметить, что все нечетные члены ряда будут получаться добавлением одного нуля после запятой в предыдущий нечетный член ряда, а все четные – добавлением одного нуля после запятой в предыдущий четный член ряда.

Рассмотрим возможные суммы первых 100 цифр после запятой для нечетных членов ряда. Для первого члена ряда это будет  $50 \cdot 10 + 50 \cdot 11 = 1050$ . Для третьего члена ряда  $50 \cdot 10 + 49 \cdot 11 = 1039$ . То есть, у нечетных членов ряда сумма может быть представлена или как  $X \cdot 21$  или как  $X \cdot 21 - 11$ , где  $X$  – натуральное число. Аналогично для четных членов ряда сумма будет равна или  $2 + Y \cdot 24 - 14 = Y \cdot 24 - 12$  или  $2 + Y \cdot 24$ . Рассмотрим число из условия –  $441_{10}$  и заметим, что это  $21 \cdot 21$ . Следовательно, это сумма цифр для нечетного члена ряда вида  $X \cdot 21$ , где  $X = 21$ . Осталось вычислить его номер. Для первого члена ряда  $X = 50$ , для пятого члена ряда  $X = 49$  и т.д. Тогда номер нужного члена ряда можно вычислить как  $(50 - X) \cdot 4 + 1 = (50 - 21) \cdot 4 + 1 = 117$

### 2. Кодирование информации. Объем информации (2 балла)

#### [Полёт]

Петя создает ПО для управления дроном. Дрон перемещается в трехмерном пространстве с заданной декартовой системой координат. Дрон умеет выполнять 6 возможных команд: вверх, вниз, вправо, влево, вперед и назад. Каждая команда перемещает дрон на 1 единицу длины в указанном направлении. Пространство не содержит никаких препятствий для перемещения дрона. Программа для дрона – это последовательность из определенного количества указанных команд. Пока дрон умеет выполнять только программы, состоящие ровно из 11 команд. Петя решил записывать программы для дрона в память в виде последовательности команд, кодируя каждую команду минимально возможным, одинаковым для всех команд количеством бит.

Вася заметил, что дрон всегда стартует из точки А и всегда заканчивает движение в точке В, смещенную на 2 единицы длины вправо, 3 единицы длины вверх и 4 единицы длины вперед. Вася сказал Пете, что тогда нет смысла выделять память исходя из необходимости хранить произвольную программу заданной длины. Он предложил перенумеровать натуральными числами все программы, которые смогут переместить дрона из точки А в точку В и сохранять в памяти номер такой программы, используя для каждого возможного номера минимально возможное, одинаковое для всех номеров количество бит. Насколько меньше бит потребуется Васе для хранения одной программы, чем предлагал Петя? В ответе укажите целое число.

**Ответ: 16**

**Решение:**

Определим объем информации, который необходимо выделить для программы, записывая её способом Пети. Для кодирования шести команд будет достаточно  $\lceil \log_2 6 \rceil = 3$  бит информации, соответственно, для 11 команд будет достаточно  $3 \cdot 11 = 33$  бит.

Определим объем информации, который необходимо выделить для программы, записывая её способом Васи. Для удобства обозначим команды буквами: вверх - U, вниз - D, вправо - R, влево - L, вперед - F, назад - B. Запишем минимальную программу, которая нужна для смещения на 2 единицы длины вправо, 3 единицы длины вверх и 4 единицы длины вперед: RRUUUFFFF. Эта программа состоит из 9 команд, но по условию в программе должно быть 11 команд, при этом итоговое перемещение дрона в пространстве должно остаться таким же. Для этого итоговая программа должна иметь вид RRUUUFFFFXY, где X и Y – две команды, выполняющие движение в противоположном направлении (либо U и D, либо R и L, либо F и B).

Посчитаем количество различных программ, которые могут быть сведены к RRUUUFFFFUD. Необходимо посчитать количество перестановок с повторениями, для этого можно либо воспользоваться готовой формулой, либо вывести формулу из тех рассуждений, что всего всего есть  $P_{11} = 11!$  перестановок 11 элементов, при этом перестановки, в которых меняются местами одинаковые буквы, мы не учитываем. Таких перестановок будет  $2! \cdot (3 + 1)! \cdot 4! \cdot 1!$ : две буквы R из минимальной программы, три буквы U из минимальной программы и ещё одна из пары UD, четыре буквы F из минимальной программы и одна буква D из пары UD. Тогда формула будет выглядеть следующим образом:

$$\frac{11!}{2! \cdot (3 + 1)! \cdot 4! \cdot 1!} = 34650$$

Итого получилось 34650 программ.

Аналогичные рассуждения можно провести для программ RRUUUFFFFRL и RRUUUFFFFFB. Для первого случая количество программ равно:

$$\frac{11!}{(2+1)! \cdot 3! \cdot 4! \cdot 1!} = 46200$$

Для второго случая количество программ равно:

$$\frac{11!}{2! \cdot 3! \cdot (4+1)! \cdot 1!} = 27720$$

И тогда на хранение номера одной программы нужно будет выделить  $\lceil \log_2(34650 + 46200 + 27720) \rceil = \lceil \log_2 108570 \rceil = 17$  бит.

Итоговый выигрыш по сравнению со способом Пети составит  $33 - 17 = 16$  бит.

### 3. Основы логики (3 балла)

#### [Кто такой Жегалкин?]

Есть логический преобразователь, на вход которому подается логическая функция от 20 переменных, не являющаяся тождественной ложью (то есть имеющая значение ИСТИНА хотя бы для одного набора значений переменных). Работает преобразователь следующим образом:

1. Строится таблица истинности данной функции.
2. Из полученной таблицы истинности берутся наборы переменных, для которых значение функции равно ИСТИНЕ.
3. Для каждого такого набора составляется полная конъюнкция всех 20 переменных, причем если значение этой переменной в наборе равно ЛЖИ, то в конъюнкции эта переменная будет с отрицанием, если значение переменной равно ИСТИНЕ, то в конъюнкции эта переменная будет без изменений.
4. Между всеми полученными полными конъюнкциями ставятся операции  $\oplus$  (Исключающее ИЛИ).
5. Все переменные с отрицанием заменяются по следующему правилу:  
 $\bar{X} = (X \oplus 1)$
6. Раскрываются все скобки по правилу:  $X \wedge (Y \oplus Z) = X \wedge Y \oplus X \wedge Z$ . После раскрытия скобок должно получиться выражение, которое может содержать некоторые отдельные переменные, конъюнкции переменных и константу ИСТИНА, соединенные Исключающим ИЛИ.
7. Из выражения убираются повторяющиеся конъюнкции. В результате у нас получается новая форма записи исходной функции. Гарантируется, что для каждой входной функции эта форма уникальна.
8. Производится проверка, остались ли в получившейся форме записи выражения операции конъюнкции. Если остались – преобразователь возвращает ЛОЖЬ, если нет – возвращает ИСТИНА.

Сколько существует таких функций от 20 переменных, которые можно подать на вход преобразователю и для которых он вернет ИСТИНУ? В ответе укажите целое число.

**Ответ: 2097151**

**Решение:**

Разберем работу алгоритма. Выполнив пункты 1-3 мы получаем набор полных конъюнкций, которые при сложении дают СДНФ исходной функции (то есть дизъюнкцию конъюнкций, состоящих из полного набора переменных, от которых зависит исходная функция). Только на шаге 4 вместо ожидаемых дизъюнкций полные конъюнкции объединяются операцией Исключающее ИЛИ. Следовательно получается запись, вида  $A \oplus B \oplus C \dots$ , где  $A, B, C \dots$  – полные конъюнкции, в которых могут присутствовать инвертированные переменные.

Проанализируем, что будет, если все инвертированные переменные заменяются на  $(a \oplus 1)$  и по указанным правилам раскрываются скобки. Если в конъюнкции присутствовала одна инвертированная переменная, то  $X(a \oplus 1) = Xa \oplus X$ . Если в конъюнкции присутствовали две инвертированные переменные, то

$$X(a \oplus 1)(b \oplus 1) = (Xa \oplus X)(b \oplus 1) = (Xa \oplus X)b \oplus 1(Xa \oplus X) = Xab \oplus Xb \oplus Xa \oplus X$$

И так далее, если посмотреть, можно заметить, что после раскрытия скобок мы получаем форму записи вида  $A \oplus B \oplus C \oplus \dots$ , где  $A, B, C, \dots$  – конъюнкции, в которых уже не обязательно присутствуют все переменные, так же важно отметить, что в них не будет инвертированных переменных, и эта конъюнкция может быть из одной переменной (либо какая то переменная, от которой зависит исходная функция, либо константа 1). Так же после избавления от пар повторяющихся конъюнкций, понимаем, что  $A, B, C \dots$  будут уникальными конъюнкциями.

Посчитаем, сколько у нас может быть функций от 20 переменных. Помним, что тождественная ложь нам не подходит по условию задачи. Тогда функций получается:  $2^{(2^{20})} - 1$

Посчитаем, сколько у нас всего может получиться форм записи, полученных после 7-го пункта алгоритма. В каждую конъюнкцию каждая переменная у нас может как входить, так и не входить (если в конъюнкцию не входит ни одна переменная – это соответствует случаю, когда конъюнкция представляет из себя константу 1). Следовательно, всего вариантов возможных конъюнкций  $2^{20}$ . Далее каждая такая конъюнкция может как входить, так и не входить в конечную форму записи, при этом хотя бы одна конъюнкция в конечной форме должна быть  $\Rightarrow$  всего таких форм у нас  $2^{(2^{20})} - 1$ .

По условию задачи, мы знаем, что для каждой входной функции получается своя уникальная форма записи. Так же мы получили, что количество входных функций и количество возможных форм записи одинаковы, следовательно, у каждой входной функции есть единственная и уникальная форма записи, представленная в алгоритме. Тогда, нам достаточно посчитать те формы, в которых конъюнкции представляют из себя одну переменную: всего таких конъюнкций будет  $20+1=21$  (20 переменных от которых зависит исходная функция и константа 1). Каждую такую конъюнкцию мы можем как включать, так и не включать в запись, но хотя бы одна конъюнкция должна присутствовать  $\Rightarrow$  всего удовлетворяющих нас форм записи (а значит и функций)  $= 2^{(n+1)} - 1 = 2^{21} - 1$ .

### 4. Кодирование информации. Алгоритмы обработки кодированной информации (1 балл)

#### [Разбегайся!]

Пусть  $N$  – шестизначное десятичное число. Последовательность цифр  $R$  получается по следующему алгоритму:

1. 1000 раз повторяется первая (самая левая) цифра числа  $N$ .

2. Цикл для всех последующих цифр числа N, двигаясь слева направо:

После каждого элемента последовательности, полученной на предыдущем шаге цикла вставить два раза подряд эту цифру

Например, если  $N=123456$ , то первые 10 элементов последовательности R, получившейся после завершения алгоритма будут:

1, 6, 6, 5, 6, 6, 5, 6, 6, 4

Известно, что в получившейся последовательности элемент с номером 209161 равен 7, элемент с номером 242758 равен 3, а элемент с номером 189890 равен 1. Нумерация элементов слева направо с 1. Определите максимальное число N, при котором это возможно, и запишите его в ответ. Если такое число невозможно, запишите в ответ NULL.

**Ответ: 399791**

**Решение:**

Пусть исходное число: 123456, то есть его цифры соответствуют их номерам считая слева направо от 1.

Рассмотрим начало последовательности после каждого шага цикла:

122122122122...

133233233133...

Обратим внимание, что после каждого шага последовательность будет состоять из одинаковых подпоследовательностей, начинающихся с каждой из тысячи единиц. Также легко заметить, что длина такой подпоследовательности будет утраиваться на каждом шаге, а, значит, после вставки последней цифры (6) длина каждой подпоследовательности, начинающейся с 1 будет  $3^5=243$ .

Рассмотрим элемент результирующей последовательности с номером 209161. Поделим номер на 243 с остатком. Остаток будет равен 181. То есть, это 181-й элемент подпоследовательности, начинающейся с 1. Обратим внимание, что в такой подпоследовательности, полученной на определенном шаге цикла, на всех позициях, остаток от деления номера которых на 3 равен 0 или 2 стоят цифры, которые вставлены на этом шаге. Но остаток от деления 181 на 3 равен 1. Значит, это не может быть последняя цифра исходного числа.

При переходе на следующий шаг цикла, номер элемента, который был в последовательности на предыдущем шаге (n), будет давать новый номер  $n*3-2$ . Поэтому символ на 181 позиции в предыдущей подпоследовательности стоял на позиции  $(181+2)/3=61$ . Остаток от деления 61 на 3 равен опять 1. Значит это не цифра, стоящая на пятой позиции в исходном числе. Символ на 61 позиции в предыдущей подпоследовательности стоял на позиции  $(61+2)/3=21$ . 21 нацело делится на 3. Следовательно, мы знаем, что это цифра, которая в исходном числе стоит на позиции 4. Таким образом, мы определили, что в исходном числе на позиции 4 стоит цифра 7.

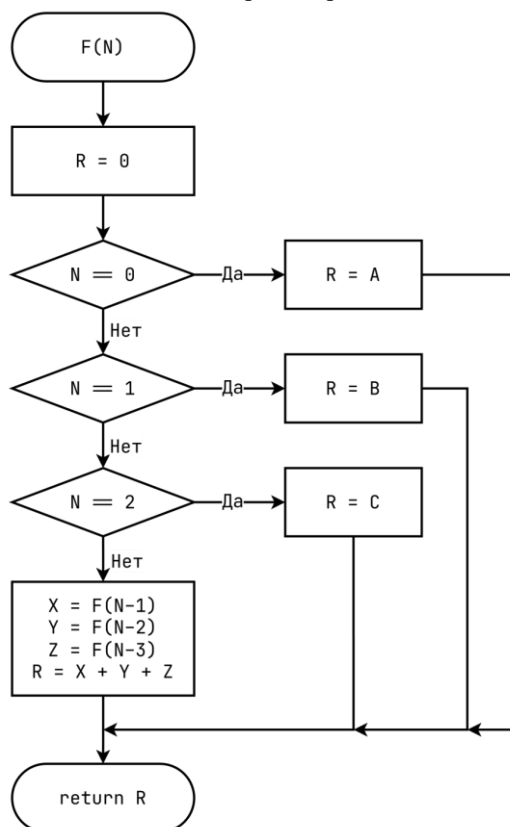
Аналогичными рассуждениями мы получим, что на позиции 1 в исходном числе стоит цифра 3 (остаток от деления 242758 на 243 равен 1), а на позиции 6 стоит цифра 1 (остаток от деления 189890 на 243 равен 107, а остаток от деления 107 на 3 равен 2). Поскольку нас просят найти максимальное число, то остальные его цифры должны быть максимальными, то есть равны 9.

Значит, это число 399791.

## 5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (1 балл)

**[Непизанские кролики]**

Дана блок-схема алгоритма, реализованного в виде рекурсивно вызываемой функции:



Известно, что  $A$ ,  $B$  и  $C$  - натуральные числа и что  $A < B < C$ .

Петя вызывает эту функцию, передавая ей в качестве входного параметра натуральное число. Известно, что для некоторого  $k$  функция возвращает следующие значения:

$$F(k) = 50890368413;$$

$$F(k+1) = 93601980590;$$

$$F(k+2) = 172160883161.$$

Определите значения  $A$ ,  $B$  и  $C$ , при которых это возможно. Если таких вариантов несколько, выберите вариант с наименьшей суммой  $A$ ,  $B$  и  $C$ . В ответе введите в указанном порядке значения  $A$ ,  $B$  и  $C$ , разделённые пробелом.

**Ответ: 1 2 5**

**Решение:**

На блок-схеме изображён модифицированный алгоритм для генерации  $N$ -го числа Фибоначчи: в нём даются первые три числа, а следующие числа определяются как сумма трёх предыдущих (при  $A=0$ ,  $B=0$  и  $C=1$  такая последовательность называется числами трибоначчи). Соответственно, нужно найти такие три числа, которые сгенерируют ряд с указанными в условии элементами. Для этого можно написать программу, которая принимает числа из условия на вход и вычитает первые два из третьего, тем самым генерируя предыдущие элементы последовательности:

```
d, e, f = 50890368413, 93601980590, 172160883161
```

```
while d > 0:
    t = f - d - e
    print(t)
    d, e, f = t, d, e
```

Результат работы программы нужно изучить достаточно внимательно, ведь в конце есть лишние элементы, которые не входят в последовательность, поскольку должно соблюдаться, что  $A, B, C \in \mathbb{N}$  и  $A < B < C$ . В данном варианте окончание вывода будет таким:

```
...
15
8
5
2
1
2
-1
```

Две последние строки нас не интересуют, и в итоге  $A=1, B=2, C=5$ .

## 6. Телекоммуникационные технологии (2 балла).

**[BGP]**

Интернет состоит из отдельных крупных сетей, управляемых отдельными организациями. Между такими сетями распределяется пространство IP-адресов и устанавливается техническое и организационное взаимодействие. Несколько упрощая, можно сказать, что такие крупные сети называются автономными системами (AS). За каждой AS закрепляется IP-подсеть.

Для обмена маршрутной информацией между маршрутизаторами AS используется внешний протокол динамической маршрутизации BGP (Border Gateway Protocol). BGP выбирает наилучший маршрут для передачи пакетов на основе ряда атрибутов и метрик.

BGP-роутер при маршрутизации пакета выбирает IP-адрес следующего по маршруту маршрутизатора, который в BGP называется «соседом» (neighbor), по набору атрибутов маршрутов. В задаче для упрощения будем считать, что реализуется следующий алгоритм.

1. по адресу назначения в IP-пакете выбираются подходящие маршруты из таблицы маршрутизации по полю «сеть назначения».
2. потом из них выбирается лучший, для чего **последовательно** проверяются атрибуты маршрутов. То есть сначала проверяется первый атрибут всех подходящих маршрутов, и если у возможных маршрутов он имеет равное значение, проверяется второй атрибут и т.д.

В задаче используются следующие атрибуты в указанном порядке:

### 1. Локальное предпочтение (Local Preference)

**Local Preference** — это атрибут, который используется для выбора исходящего трафика из автономной системы. Маршрут с **наибольшим** значением Local Preference считается предпочтительным.

### 2. Наименьший Путь AS (AS Path)

BGP предпочитает маршруты с **наименьшим** количеством автономных систем в пути (AS Path). Чем короче путь, тем лучше.

### 3. Наименьший IGP Metric до Next Hop

Если есть несколько маршрутов с одинаковыми атрибутами, BGP выбирает маршрут с **наименьшим** значением метрики IGP до Next Hop (следующего прыжка).

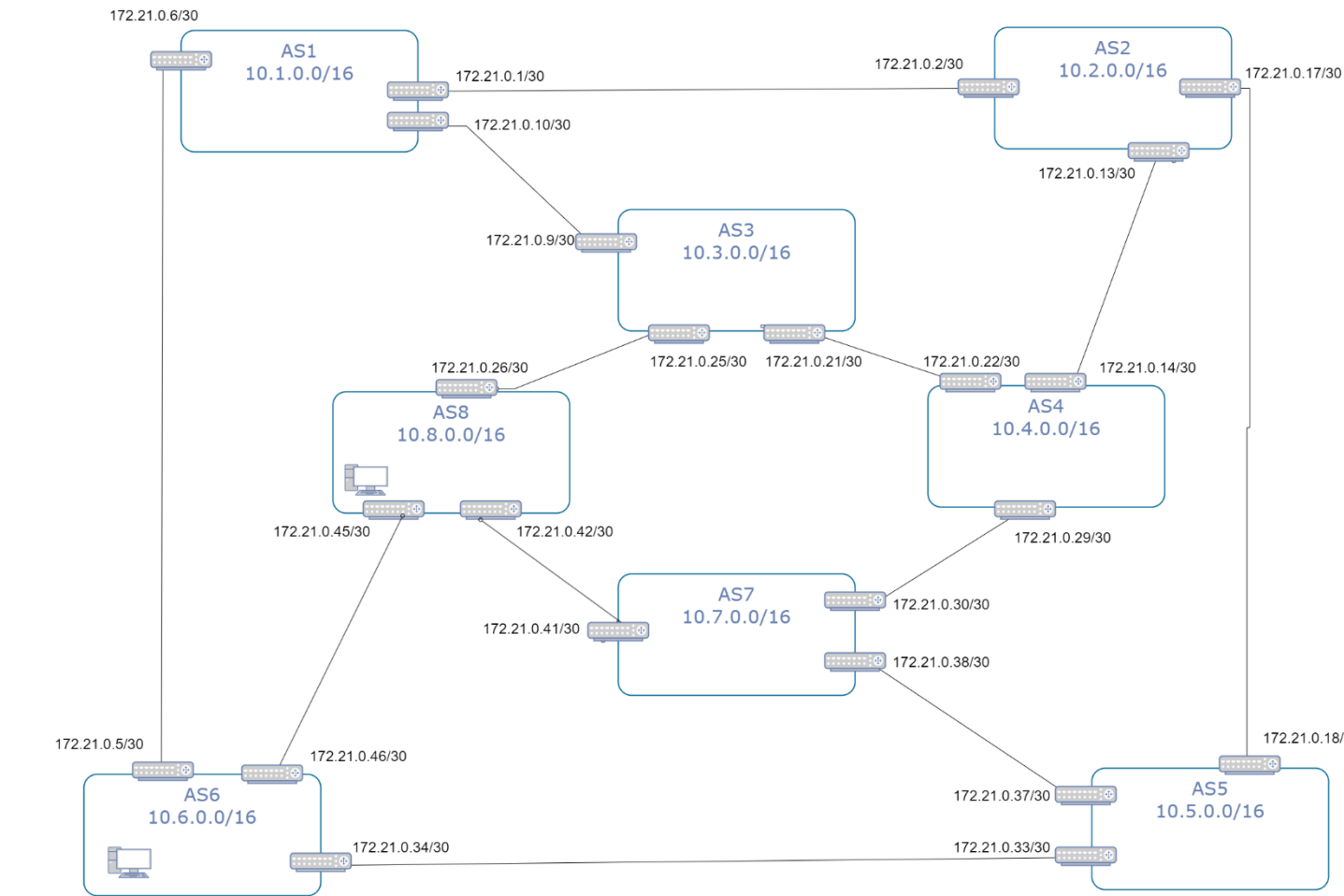
### 4. Старейший маршрут (Oldest Route)

Если все предыдущие атрибуты равны, BGP предпочитает **старейший** маршрут, так как он считается более стабильным.

### 5. Наименьший IP-адрес соседа

В крайнем случае, если все предыдущие атрибуты равны, BGP выбирает маршрут, полученный от соседа с **наименьшим** IP-адресом (сравнение производится как сравнение 32-битных значений адресов).

Далее приведены схема сети и фрагменты таблиц маршрутов для каждой из AS (сделано это для простоты, в реальности эти данные есть в каждом граничном маршрутизаторе AS).



AS1:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.5.0.0/16     | AS2                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.2   |
| 10.1.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.3.0.0/16     | AS3                | 200              | 1             | 10                     | 2025-02-03              | 172.21.0.9   |
| 10.7.0.0/16     | AS2                | 100              | 3             | 20                     | 2025-02-03              | 172.21.0.2   |
| 10.8.0.0/16     | AS3                | 100              | 2             | 20                     | 2025-02-01              | 172.21.0.9   |
| 10.6.0.0/16     | AS6                | 100              | 2             | 20                     | 2025-02-01              | 172.21.0.5   |
| 10.6.0.0/16     | AS3                | 100              | 2             | 20                     | 2025-02-01              | 172.21.0.9   |
| 10.3.0.0/16     | AS6                | 200              | 3             | 50                     | 2025-02-23              | 172.21.0.5   |
| 10.4.0.0/16     | AS3                | 200              | 2             | 5                      | 2025-02-03              | 172.21.0.9   |

AS2:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.2.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.3.0.0/16     | AS1                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.1   |
| 10.6.0.0/16     | AS1                | 100              | 2             | 20                     | 2025-02-10              | 172.21.0.1   |
| 10.7.0.0/16     | AS4                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.14  |
| 10.8.0.0/16     | AS4                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.14  |
| 10.6.0.0/16     | AS5                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.18  |
| 10.3.0.0/16     | AS4                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.14  |
| 10.4.0.0/16     | AS1                | 200              | 2             | 20                     | 2025-02-03              | 172.21.0.1   |
| 10.4.0.0/16     | AS4                | 100              | 1             | 20                     | 2025-02-03              | 172.21.0.14  |
| 10.1.0.0/16     | AS1                | 100              | 1             | 20                     | 2025-02-03              | 172.21.0.1   |
| 10.1.0.0/16     | AS4                | 200              | 3             | 20                     | 2025-02-03              | 172.21.0.14  |

## AS3:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.5.0.0/16     | AS4                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.22  |
| 10.6.0.0/16     | AS1                | 100              | 2             | 20                     | 2025-02-01              | 172.21.0.10  |
| 10.6.0.0/16     | AS4                | 200              | 4             | 10                     | 2025-02-03              | 172.21.0.22  |
| 10.7.0.0/16     | AS4                | 100              | 3             | 10                     | 2025-02-04              | 172.21.0.22  |
| 10.3.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.8.0.0/16     | AS8                | 100              | 3             | 10                     | 2025-02-04              | 172.21.0.26  |
| 10.4.0.0/16     | AS4                | 200              | 1             | 10                     | 2025-02-04              | 172.21.0.22  |
| 10.4.0.0/16     | AS8                | 100              | 3             | 20                     | 2025-02-04              | 172.21.0.26  |
| 10.1.0.0/16     | AS8                | 100              | 3             | 20                     | 2025-02-04              | 172.21.0.26  |
| 10.1.0.0/16     | AS1                | 100              | 1             | 20                     | 2025-02-04              | 172.21.0.10  |

## AS4:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.5.0.0/16     | AS2                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.13  |
| 10.6.0.0/16     | AS2                | 100              | 3             | 5                      | 2025-02-03              | 172.21.0.13  |
| 10.7.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.6.0.0/16     | AS7                | 100              | 3             | 10                     | 2025-02-01              | 172.21.0.30  |
| 10.8.0.0/16     | AS3                | 100              | 1             | 10                     | 2025-02-01              | 172.21.0.21  |
| 10.6.0.0/16     | AS3                | 100              | 3             | 5                      | 2025-02-11              | 172.21.0.21  |
| 10.1.0.0/16     | AS3                | 100              | 3             | 5                      | 2025-02-01              | 172.21.0.21  |

## AS5:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.4.0.0/16     | AS2                | 100              | 2             | 10                     | 2025-02-02              | 172.21.0.17  |
| 10.5.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.6.0.0/16     | AS6                | 100              | 1             | 20                     | 2025-02-03              | 172.21.0.34  |
| 10.8.0.0/16     | AS7                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.38  |
| 10.3.0.0/16     | AS2                | 500              | 3             | 20                     | 2025-02-01              | 172.21.0.17  |
| 10.2.0.0/16     | AS2                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.17  |
| 10.3.0.0/16     | AS6                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.34  |
| 10.4.0.0/16     | AS7                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.38  |
| 10.1.0.0/16     | AS6                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.34  |
| 10.1.0.0/16     | AS2                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.17  |

## AS6:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.5.0.0/16     | AS1                | 100              | 3             | 20                     | 2025-02-02              | 172.21.0.6   |
| 10.6.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.4.0.0/16     | AS5                | 100              | 3             | 51                     | 2025-02-11              | 172.21.0.33  |
| 10.7.0.0/16     | AS1                | 100              | 4             | 20                     | 2025-02-03              | 172.21.0.6   |
| 10.8.0.0/16     | AS1                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.6   |
| 10.3.0.0/16     | AS8                | 200              | 2             | 5                      | 2025-03-08              | 172.21.0.45  |
| 10.4.0.0/16     | AS1                | 100              | 3             | 52                     | 2025-02-11              | 172.21.0.6   |
| 10.3.0.0/16     | AS5                | 100              | 4             | 50                     | 2025-02-11              | 172.21.0.33  |
| 10.4.0.0/16     | AS8                | 100              | 3             | 50                     | 2025-02-11              | 172.21.0.45  |

## AS7:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.5.0.0/16     | AS4                | 100              | 3             | 20                     | 2025-02-02              | 172.21.0.29  |
| 10.6.0.0/16     | AS5                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.37  |
| 10.7.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.8.0.0/16     | AS4                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.29  |
| 10.4.0.0/16     | AS4                | 100              | 1             | 20                     | 2025-02-01              | 172.21.0.29  |
| 10.1.0.0/16     | AS8                | 300              | 3             | 20                     | 2025-02-01              | 172.21.0.42  |
| 10.1.0.0/16     | AS4                | 200              | 3             | 20                     | 2025-02-01              | 172.21.0.29  |
| 10.1.0.0/16     | AS5                | 400              | 5             | 80                     | 2025-02-01              | 172.21.0.37  |

AS8:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.6.0.0/16     | AS3                | 200              | 3             | 10                     | 2025-02-01              | 172.21.0.25  |
| 10.7.0.0/16     | AS3                | 200              | 3             | 10                     | 2025-02-05              | 172.21.0.25  |
| 10.5.0.0/16     | AS3                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.25  |
| 10.4.0.0/16     | AS7                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.41  |
| 10.8.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.2.0.0/16     | AS7                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.41  |
| 10.6.0.0/16     | AS7                | 200              | 3             | 20                     | 2025-02-01              | 172.21.0.41  |
| 10.3.0.0/16     | AS3                | 200              | 1             | 10                     | 2025-02-15              | 172.21.0.25  |
| 10.4.0.0/16     | AS3                | 100              | 2             | 10                     | 2025-02-03              | 172.21.0.25  |
| 10.6.0.0/16     | AS6                | 50               | 1             | 20                     | 2025-02-03              | 172.21.0.46  |

Определите сначала маршрут прохождения пакета из AS8 в AS6, а потом маршрут прохождения пакета в случае, если из-за аварии будет отключены связи между AS8 и AS3, а также между AS1 и AS6, а записи, соответствующие этим связям, будут исключены из таблиц.

В ответ запишите номера автономных систем через запятую, начиная с 8 и заканчивая 6, сначала для первого случая, а потом, через тире, для второго. Если маршрут построить нельзя (например, его нет или образуется цикл), вместо перечисления номеров автономных систем укажите NULL.

Пример ввода ответа: 8,10,12,6-8,10,1,2,6.

**Ответ: 8,3,4,2,5,6-8,7,5,6**

**Решение:**

В данной задаче необходимо аккуратно пройти по таблицам маршрутизации и последовательно выбирать нужные строки.

Поскольку маршрут начинается в AS8, посмотрим таблицу маршрутизации этой автономной системы. В ней нужно выбрать те маршруты, в которых сеть назначения равна 10.6.0.0/16. Строки с такими маршрутами выделим зелёным цветом:

| Сеть назначения    | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа       |
|--------------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------------|
| <b>10.6.0.0/16</b> | <b>AS3</b>         | <b>200</b>       | <b>3</b>      | <b>10</b>              | <b>2025-02-01</b>       | <b>172.21.0.25</b> |
| 10.7.0.0/16        | AS3                | 200              | 3             | 10                     | 2025-02-05              | 172.21.0.25        |
| 10.5.0.0/16        | AS3                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.25        |
| 10.4.0.0/16        | AS7                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.41        |
| 10.8.0.0/16        | Локально           | 500              | 0             | 0                      | 2025-02-01              | -                  |
| 10.2.0.0/16        | AS7                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.41        |
| <b>10.6.0.0/16</b> | <b>AS7</b>         | <b>200</b>       | <b>3</b>      | <b>20</b>              | <b>2025-02-01</b>       | <b>172.21.0.41</b> |
| 10.3.0.0/16        | AS3                | 200              | 1             | 10                     | 2025-02-15              | 172.21.0.25        |
| 10.4.0.0/16        | AS3                | 100              | 2             | 10                     | 2025-02-03              | 172.21.0.25        |
| <b>10.6.0.0/16</b> | <b>AS6</b>         | <b>50</b>        | <b>1</b>      | <b>20</b>              | <b>2025-02-03</b>       | <b>172.21.0.46</b> |

При сравнении Local Preference из рассмотрения убирается маршрут в AS6, так как там значение меньше, а дальше при сравнении IGP Metric убирается маршрут в AS7, поскольку там значение больше. Остаётся маршрут в AS3.

Дальше аналогично рассмотрим маршруты из AS3:

| Сеть назначения    | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа       |
|--------------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------------|
| 10.5.0.0/16        | AS4                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.22        |
| <b>10.6.0.0/16</b> | <b>AS1</b>         | <b>100</b>       | <b>2</b>      | <b>20</b>              | <b>2025-02-01</b>       | <b>172.21.0.10</b> |
| <b>10.6.0.0/16</b> | <b>AS4</b>         | <b>200</b>       | <b>4</b>      | <b>10</b>              | <b>2025-02-03</b>       | <b>172.21.0.22</b> |
| 10.7.0.0/16        | AS4                | 100              | 3             | 10                     | 2025-02-04              | 172.21.0.22        |
| 10.3.0.0/16        | Локально           | 500              | 0             | 0                      | 2025-02-01              | -                  |
| 10.8.0.0/16        | AS8                | 100              | 3             | 10                     | 2025-02-04              | 172.21.0.26        |
| 10.4.0.0/16        | AS4                | 200              | 1             | 10                     | 2025-02-04              | 172.21.0.22        |
| 10.4.0.0/16        | AS8                | 100              | 3             | 20                     | 2025-02-04              | 172.21.0.26        |
| 10.1.0.0/16        | AS8                | 100              | 3             | 20                     | 2025-02-04              | 172.21.0.26        |
| 10.1.0.0/16        | AS1                | 100              | 1             | 20                     | 2025-02-04              | 172.21.0.10        |

При сравнении Local Preference из рассмотрения убирается маршрут в AS1, и остаётся маршрут в AS4.

Рассмотрим маршруты из AS4:

| Сеть назначения    | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа       |
|--------------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------------|
| 10.5.0.0/16        | AS2                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.13        |
| <b>10.6.0.0/16</b> | <b>AS2</b>         | <b>100</b>       | <b>3</b>      | <b>5</b>               | <b>2025-02-03</b>       | <b>172.21.0.13</b> |
| 10.7.0.0/16        | Локально           | 500              | 0             | 0                      | 2025-02-01              | -                  |
| <b>10.6.0.0/16</b> | <b>AS7</b>         | <b>100</b>       | <b>3</b>      | <b>10</b>              | <b>2025-02-01</b>       | <b>172.21.0.30</b> |
| 10.8.0.0/16        | AS3                | 100              | 1             | 10                     | 2025-02-01              | 172.21.0.21        |
| <b>10.6.0.0/16</b> | <b>AS3</b>         | <b>100</b>       | <b>3</b>      | <b>5</b>               | <b>2025-02-11</b>       | <b>172.21.0.21</b> |
| 10.1.0.0/16        | AS3                | 100              | 3             | 5                      | 2025-02-01              | 172.21.0.21        |

При сравнении IGP Metric из рассмотрения убирается маршрут в AS7 (до этого все метрики равны), при сравнении даты появления маршрута из рассмотрения также убирается маршрут в AS3, так как он появился позже. Остаётся маршрут в AS2.

Рассмотрим маршруты из AS2:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.2.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.3.0.0/16     | AS1                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.1   |
| 10.6.0.0/16     | AS1                | 100              | 2             | 20                     | 2025-02-10              | 172.21.0.1   |
| 10.7.0.0/16     | AS4                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.14  |
| 10.8.0.0/16     | AS4                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.14  |
| 10.6.0.0/16     | AS5                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.18  |
| 10.3.0.0/16     | AS4                | 100              | 2             | 20                     | 2025-02-03              | 172.21.0.14  |
| 10.4.0.0/16     | AS1                | 200              | 2             | 20                     | 2025-02-03              | 172.21.0.1   |
| 10.4.0.0/16     | AS4                | 100              | 1             | 20                     | 2025-02-03              | 172.21.0.14  |
| 10.1.0.0/16     | AS1                | 100              | 1             | 20                     | 2025-02-03              | 172.21.0.1   |
| 10.1.0.0/16     | AS4                | 200              | 3             | 20                     | 2025-02-03              | 172.21.0.14  |

У двух нужных маршрутов все метрики до даты появления совпадают, а маршрут в AS5 старше маршрута в AS1, поэтому маршрут в AS1 убирается из рассмотрения, и остаётся маршрут в AS5.

Рассмотрим маршруты из AS5:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.4.0.0/16     | AS2                | 100              | 2             | 10                     | 2025-02-02              | 172.21.0.17  |
| 10.5.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.6.0.0/16     | AS6                | 100              | 1             | 20                     | 2025-02-03              | 172.21.0.34  |
| 10.8.0.0/16     | AS7                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.38  |
| 10.3.0.0/16     | AS2                | 500              | 3             | 20                     | 2025-02-01              | 172.21.0.17  |
| 10.2.0.0/16     | AS2                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.17  |
| 10.3.0.0/16     | AS6                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.34  |
| 10.4.0.0/16     | AS7                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.38  |
| 10.1.0.0/16     | AS6                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.34  |
| 10.1.0.0/16     | AS2                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.17  |

В нём только один маршрут ведёт в нужную сеть, поэтому рассмотрим маршруты из AS6:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.5.0.0/16     | AS1                | 100              | 3             | 20                     | 2025-02-02              | 172.21.0.6   |
| 10.6.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.4.0.0/16     | AS5                | 100              | 3             | 51                     | 2025-02-11              | 172.21.0.33  |
| 10.7.0.0/16     | AS1                | 100              | 4             | 20                     | 2025-02-03              | 172.21.0.6   |
| 10.8.0.0/16     | AS1                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.6   |
| 10.3.0.0/16     | AS8                | 200              | 2             | 5                      | 2025-03-08              | 172.21.0.45  |
| 10.4.0.0/16     | AS1                | 100              | 3             | 52                     | 2025-02-11              | 172.21.0.6   |
| 10.3.0.0/16     | AS5                | 100              | 4             | 50                     | 2025-02-11              | 172.21.0.33  |
| 10.4.0.0/16     | AS8                | 100              | 3             | 50                     | 2025-02-11              | 172.21.0.45  |

Здесь запрос в сеть назначения обрабатывается локально, т.е. текущей AS. Соответственно, пакет доставлен в AS6.

Итоговый маршрут получается следующим: **8,3,4,2,5,6**.

Дальше нужно так же пройти по таблицам с учётом ситуации, когда из строя выходят каналы связи AS8 – AS3 и AS1 – AS6. Поскольку каналы связи двухсторонние, из таблиц маршрутизации удаляются записи не только из AS8 в AS3 и из AS1 в AS6, но и наоборот. Это сильно повлияет на маршрут пакета, поскольку исходно в первый раз пакет передаётся как раз из AS8 в AS3.

Рассмотрим маршруты из AS3 с учётом ограничений. Красным цветом выделены недоступные маршруты:

| Сеть назначения | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|-----------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.6.0.0/16     | AS3                | 200              | 3             | 10                     | 2025-02-01              | 172.21.0.25  |
| 10.7.0.0/16     | AS3                | 200              | 3             | 10                     | 2025-02-05              | 172.21.0.25  |
| 10.5.0.0/16     | AS3                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.25  |
| 10.4.0.0/16     | AS7                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.41  |
| 10.8.0.0/16     | Локально           | 500              | 0             | 0                      | 2025-02-01              | -            |
| 10.2.0.0/16     | AS7                | 200              | 2             | 10                     | 2025-02-03              | 172.21.0.41  |
| 10.6.0.0/16     | AS7                | 200              | 3             | 20                     | 2025-02-01              | 172.21.0.41  |
| 10.3.0.0/16     | AS3                | 200              | 1             | 10                     | 2025-02-15              | 172.21.0.25  |
| 10.4.0.0/16     | AS3                | 100              | 2             | 10                     | 2025-02-03              | 172.21.0.25  |
| 10.6.0.0/16     | AS6                | 50               | 1             | 20                     | 2025-02-03              | 172.21.0.46  |

Из двух оставшихся маршрутов, которые подходят, нужно выбрать маршрут в AS7, поскольку его значение Local Preference больше, чем у маршрута в AS6.

Рассмотрим маршруты из AS7:

| Сеть назначения    | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа       |
|--------------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------------|
| 10.5.0.0/16        | AS4                | 100              | 3             | 20                     | 2025-02-02              | 172.21.0.29        |
| <b>10.6.0.0/16</b> | <b>AS5</b>         | <b>100</b>       | <b>2</b>      | <b>20</b>              | <b>2025-02-03</b>       | <b>172.21.0.37</b> |
| 10.7.0.0/16        | Локально           | 500              | 0             | 0                      | 2025-02-01              | -                  |
| 10.8.0.0/16        | AS4                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.29        |
| 10.4.0.0/16        | AS4                | 100              | 1             | 20                     | 2025-02-01              | 172.21.0.29        |
| 10.1.0.0/16        | AS8                | 300              | 3             | 20                     | 2025-02-01              | 172.21.0.42        |
| 10.1.0.0/16        | AS4                | 200              | 3             | 20                     | 2025-02-01              | 172.21.0.29        |
| 10.1.0.0/16        | AS5                | 400              | 5             | 80                     | 2025-02-01              | 172.21.0.37        |

Здесь подходит только один маршрут, в AS5. Рассмотрим маршруты из AS5:

| Сеть назначения    | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа       |
|--------------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------------|
| 10.4.0.0/16        | AS2                | 100              | 2             | 10                     | 2025-02-02              | 172.21.0.17        |
| 10.5.0.0/16        | Локально           | 500              | 0             | 0                      | 2025-02-01              | -                  |
| <b>10.6.0.0/16</b> | <b>AS6</b>         | <b>100</b>       | <b>1</b>      | <b>20</b>              | <b>2025-02-03</b>       | <b>172.21.0.34</b> |
| 10.8.0.0/16        | AS7                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.38        |
| 10.3.0.0/16        | AS2                | 500              | 3             | 20                     | 2025-02-01              | 172.21.0.17        |
| 10.2.0.0/16        | AS2                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.17        |
| 10.3.0.0/16        | AS6                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.34        |
| 10.4.0.0/16        | AS7                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.38        |
| 10.1.0.0/16        | AS6                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.34        |
| 10.1.0.0/16        | AS2                | 100              | 2             | 20                     | 2025-02-02              | 172.21.0.17        |

Таблица аналогична таблице в прошлом случае, поэтому рассмотрим маршруты из AS6:

| Сеть назначения    | Номер следующей AS | Local Preference | Длина AS Path | IGP Metric до Next Hop | Дата появления маршрута | Адрес соседа |
|--------------------|--------------------|------------------|---------------|------------------------|-------------------------|--------------|
| 10.5.0.0/16        | AS1                | 100              | 3             | 20                     | 2025-02-02              | 172.21.0.6   |
| <b>10.6.0.0/16</b> | <b>Локально</b>    | <b>500</b>       | <b>0</b>      | <b>0</b>               | <b>2025-02-01</b>       | <b>-</b>     |
| 10.4.0.0/16        | AS5                | 100              | 3             | 51                     | 2025-02-11              | 172.21.0.33  |
| 10.7.0.0/16        | AS1                | 100              | 4             | 20                     | 2025-02-03              | 172.21.0.6   |
| 10.8.0.0/16        | AS1                | 100              | 3             | 20                     | 2025-02-01              | 172.21.0.6   |
| 10.3.0.0/16        | AS8                | 200              | 2             | 5                      | 2025-03-08              | 172.21.0.45  |
| 10.4.0.0/16        | AS1                | 100              | 3             | 52                     | 2025-02-11              | 172.21.0.6   |
| 10.3.0.0/16        | AS5                | 100              | 4             | 50                     | 2025-02-11              | 172.21.0.33  |
| 10.4.0.0/16        | AS8                | 100              | 3             | 50                     | 2025-02-11              | 172.21.0.45  |

Таблица, хоть и содержит удалённые маршруты, всё равно имеет только один маршрут для пакетов в нужную сеть, причём локальный, то есть пакет доставлен в AS6.

Итоговый маршрут получается следующим: **8,7,5,6**.

Ответ к задаче будет записываться следующим образом: **8,3,4,2,5,6-8,7,5,6**.

## 7. Технологии сортировки и фильтрации данных (2 балл)

### [Космический капитализм]

В свободное от работы время Николай Владимирович, ответственный за систему проведения олимпиад, разрабатывает новый космический симулятор, и на данный момент у него готова система инвентаря корабля с оборудованием, гиперпрыжки между двумя звёздными системами и торговля на планетах. Рассмотрим подробнее устройство симулятора.

Игровой мир состоит из звёздных систем, в каждой из которых есть несколько планет. Для перемещения корабля между звёздными системами необходим двигатель и топливный бак. На один гиперпрыжок корабль тратит количество топлива, равное расстоянию между звёздными системами в парсеках. Из этого следует, что корабль не может выполнить прыжок на расстояние больше, чем вместимость его топливного бака.

Владелец корабля имеет определённое количество галактических кредитов на счёте. На каждой планете представлены товары пяти видов: продовольствие, медикаменты, техника, роскошь и минералы. Для каждой планеты и каждого вида товара известно количество имеющегося товара в тоннах, стоимость покупки и продажи в галактических кредитах. Корпус корабля имеет заданную грузоподъёмность, и нельзя купить товаров больше, чем имеющаяся грузоподъёмность.

Сейчас Николай Владимирович пишет тест для проверки этой функциональности. Тестовый сценарий состоит из нескольких шагов:

1. Приобрести товар **одного вида** на текущей планете в некотором количестве;
2. Взлететь с планеты и выполнить гиперпрыжок **в другую систему**;
3. Приземлиться на выбранной планете и продать весь купленный на шаге 1 товар.

Для этого сценария есть отдельный файл сохранения, для которого известно количество товара на каждой планете, стоимость его покупки и продажи. Эти данные представлены в виде отдельной базы данных, которая доступна ниже.

Таблица stars хранит информацию о звёздных системах:

- id — уникальный идентификатор системы в рамках игрового мира;
- name — название системы.

Таблица planets хранит информацию о планетах:

- id — уникальный идентификатор планеты в рамках игрового мира;
- name — название планеты;
- star\_id — идентификатор звёздной системы, к которой относится планета.

Таблица star\_map хранит информацию о расстояниях между звёздными системами:

- star\_from\_id — идентификатор начальной звёздной системы;
- star\_to\_id — идентификатор звёздной системы назначения (обязательно отличается от star\_from\_id);
- distance — расстояние между этими звёздными системами в парсеках.

Таблица product\_types хранит информацию о типах товаров:

- id — уникальный идентификатор типа товаров;
- name — название типа товаров.

Таблица prices хранит информацию о стоимости покупки и продажи каждого типа товаров на каждой из планет:

- planet\_id — идентификатор планеты;
- product\_type — идентификатор типа товаров;
- buy\_price — стоимость покупки владельцем корабля одной тонны товара в галактических кредитах;
- sell\_price — стоимость продажи владельцем корабля одной тонны товара в галактических кредитах;
- amount — количество товара в тоннах, доступное к покупке на данной планете.

В рамках данного сценария кредитов хватит для покупки любого количества любого товара на любой планете. Пока что в этом сценарии не определено, какой товар и в каком количестве нужно приобрести и на какой планете его продать. Николай Владимирович хочет подобрать эти параметры так, чтобы максимизировать полученную прибыль (разницу между стоимостью продажи и покупки). Известно, что в сохранении корабль находится на планете Раалито системы Хезе, грузоподъёмность корабля равна 80 тонн, а вместимости топливного бака хватит на прыжок расстоянием в 25 парсек. Как опытный программист, Николай Владимирович определил нужные параметры за несколько минут и дописал тест. Определите это вместе с ним и вы. В качестве ответа введите одно целое число — прибыль в галактических кредитах, полученную после выполнения тестового сценария с найденными параметрами. Если окажется, что для тестового сценария нет ни одного набора параметров, который бы дал положительную прибыль, в качестве ответа введите 0.

**Ответ: 6080**

**Решение:**

Чтобы не писать длинный запрос с подзапросами, разобьём решение на три части.

Для начала нужно узнать идентификаторы планеты и соответствующей планетарной системы. Это можно выполнить одним запросом (заодно убедимся, что условие верное, и планета действительно находится в указанной системе):

```
SELECT p.id, p.name, s.id, s.name FROM planets p
INNER JOIN main.stars s ON s.id = p.star_id
WHERE p.name = 'Раалито';
```

Получится следующий вывод:

| p.id | p.name  | s.id | s.name |
|------|---------|------|--------|
| 54   | Раалито | 10   | Хезе   |

Теперь определим все планеты, которые находятся в системах, удалённых от текущей не более чем на 25 парсек:

```
SELECT p.* FROM star_map sm
INNER JOIN planets p ON star_to_id = p.star_id
WHERE star_from_id = 10 AND distance <= 25;
```

Получится следующая таблица:

| id  | name           | star_id |
|-----|----------------|---------|
| 1   | Арнарик Гаудад | 1       |
| 2   | Рамгатру       | 1       |
| 3   | Хартис         | 1       |
| 4   | Проту Памперой | 1       |
| 5   | Калайна        | 1       |
| 6   | Унашера        | 1       |
| 81  | Гешши          | 17      |
| 82  | Халгалла       | 17      |
| 83  | Ципца          | 17      |
| 84  | Эйя            | 17      |
| 85  | Суджим         | 17      |
| 103 | Оннд           | 22      |
| 104 | Ушалла         | 22      |
| 105 | Индерон        | 22      |
| 106 | Энжа           | 22      |
| 107 | Гаших          | 22      |

|     |            |    |
|-----|------------|----|
| 108 | Уррагах    | 22 |
| 133 | Зэмин      | 28 |
| 134 | Рэдэа      | 28 |
| 135 | Пунэ-анита | 28 |
| 136 | Хуку       | 28 |
| 167 | Перегон    | 37 |
| 168 | Глоткус    | 37 |
| 169 | Мабэл      | 37 |
| 170 | Дарзания   | 37 |
| 171 | Ойкнур Бад | 37 |
| 183 | Альянони   | 40 |
| 184 | Горик      | 40 |
| 185 | Эгмер      | 40 |
| 186 | Итниклей   | 40 |
| 299 | Локон      | 66 |
| 300 | Га-по      | 66 |
| 301 | Иоханг     | 66 |
| 302 | Шолоани    | 66 |
| 303 | Пунисе     | 66 |

Третий запрос позволит построить таблицу с возможной прибылью от продажи того или иного товара на той или иной планете. Здесь необходимо вспомнить про ограничение грузоподъёмности корабля в 80 тонн:

```

SELECT pr.planet_id, pr.product_type, pr.sell_price, pr_from.buy_price,
pr_from.amount,
      MIN(pr_from.amount, 80) * (pr.sell_price - pr_from.buy_price) profit
FROM prices pr
INNER JOIN prices pr_from ON pr_from.planet_id = 54 AND pr_from.product_type =
pr.product_type
WHERE pr.planet_id IN (1, 2, 3, 4, 5, 6, 81, 82, 83, 84, 85, 103, 104, 105, 106, 107,
108, 133, 134, 135, 136, 167, 168, 169, 170, 171, 183, 184, 185, 186, 299, 300, 301,
302, 303)
ORDER BY profit DESC;

```

Первые 10 строк результата будут выглядеть так:

| pr.planet_id | pr.product_type | pr.sell_price | pr_from.buy_price | pr_from.amount | profit |
|--------------|-----------------|---------------|-------------------|----------------|--------|
| 103          | 1               | 142           | 66                | 865            | 6080   |
| 183          | 1               | 136           | 66                | 865            | 5600   |
| 301          | 1               | 134           | 66                | 865            | 5440   |
| 186          | 1               | 133           | 66                | 865            | 5360   |
| 135          | 1               | 132           | 66                | 865            | 5280   |
| 133          | 1               | 129           | 66                | 865            | 5040   |
| 81           | 1               | 128           | 66                | 865            | 4960   |
| 83           | 1               | 128           | 66                | 865            | 4960   |
| 300          | 1               | 126           | 66                | 865            | 4800   |
| 82           | 1               | 125           | 66                | 865            | 4720   |

Получается, что максимальная прибыль равна 6080 галактических кредитов. Её можно получить, если купить 80 тонн минералов и продать их на планете Оннд.

Также можно было написать простые запросы вида **SELECT \* FROM** [название таблицы] для всех указанных в условии таблиц и обработать данные вручную или воспользоваться библиотекой для доступа к SQL БД для используемого языка программирования, скачав представленный в условии файл с базой данных.

### **Задание 8 (3 балла) и Задание 9 (4 балла). Технологии программирования**

В отдельном архиве опубликованы тесты, решения жюри и проверяющие программы для задач по технологиям программирования. Архив содержит 4 директории: по два варианта каждой из двух задач. В поддиректории tests каждой задачи содержатся тесты и ответы решения жюри в файлах "???" и "???.a" соответственно, где "???" -- двузначное число с, возможно, ведущим нулем. В поддиректории solutions каждой задачи содержатся решения жюри на различных языках программирования, а также примеры неверных решений (см. файлы с расширением \*.desc, чтобы классифицировать решения). В корневой директории архива находится программа, использовавшаяся для проверки ответа участника (check.cpp). В поддиректории statements содержатся условия задач. Апелляции заданий по программированию должны быть основаны на предоставленных тестах.

Наименования директорий с задачами (по вариантам): задача 8 – jobs-scheduling-v1, jobs-scheduling-v2; задача 9 – tournament-rewards-v1, tournament-rewards-v2.

## Расписание станков

Это по большей части задача на реализацию, требующая при этом воспользоваться идеей ленивых вычислений. Разберем ее по частям.

Начнем с того, что, собственно, требуется хранить:

1. Для каждой задачи необходимо знать время начала и завершения ее обработки. Поскольку эти значения могут быть выставлены как последовательно при обработке, так и в любой момент времени при отмене, имеет смысл просто иметь доступ к задаче по ее номеру за  $O(1)$ : заведем словарь (unordered map, hash map) `jobs`, сопоставляющий номера задач и сами задачи.
2. Также для задачи полезно понимать, срочная ли она (`job.is_urgent`), и какой прогресс по ней уже выполнен (`job.progress`). Ну и саму длительность задачи `job.duration` тоже запоним отдельным полем. Определим `job.remaining()` как 0, если она отменена, и как разницу между длительностью и прогрессом иначе.
3. Для каждого станка в любой момент времени может понадобиться узнать время его остановки. Мы будем хранить это отдельным полем `machine.finish` для каждого станка, так как это самый эффективный способ быстро пересчитывать это значение при добавлении или отмене задач.
4. Разумеется, для каждого станка надо хранить указанную в условии очередь задач `machine.queue`.
5. И для обновления `machine.finish` надо хранить словарь, сопоставляющий задачи станкам, на которых они обрабатываются.

Однако этого недостаточно. В этой задаче необходимы ленивые вычисления по одной причине: отмененная задача может находиться в середине очереди, из-за чего ее явное удаление оттуда может быть достаточно долгой операцией. Можно было бы в качестве альтернативы хранить упорядоченное множество вместо очереди, однако это замедлит и усложнит решение.

Идея ленивых вычислений в следующем: вместо реального удаления задач будем просто пометить задачу удаленной (`job.is_cancelled`), а удалим ее из очереди, когда очередь до нее дойдет. Вторая идея — обрабатывать задачи не в режиме реального времени, а по необходимости. Действительно, станки независимы, поэтому можно проделать всю работу станка между предыдущим и текущим моментом времени только тогда, когда есть необходимость, например, понять где сейчас начало очереди. Поэтому будем дополнительно для каждого станка хранить последний момент времени, на котором мы «остановились» `machine.t`.

Тогда заведем следующие методы:

```
func (machine) work_until(time):
    while machine.t < time and len(machine.queue) > 0:
        job = machine.queue[0]
        if job.is_cancelled:
            machine.queue.popleft()
            continue

        machine.process_job(job, time)
        if job.remaining() == 0:
            machine.queue.popleft()

    machine.t = time
```

Иными словами, при необходимости в момент времени  $t$  узнать актуальное состояние станка, выполним всю его работу до этого момента: будем увеличивать прогресс по работам по очереди, пока не дойдем до текущего момента. При этом `process_job` выглядит следующим образом:

```

func (machine) process_job(job, time):
    rem = min(time - machine.t, job.remaining())
    if job.start is None:
        job.start = machine.t

    job.progress += rem
    machine.t += rem
    if job.remaining():
        job.finish = machine.t

```

Здесь мы увеличиваем прогресс по задаче на минимум из того, сколько осталось времени до ее завершения и до достижения момента `time`. Не забываем при этом выставить начало и конец обработки задачи в соответствии с определениями из условия.

Остается только аккуратно обработать все запросы. Перед выполнением запроса к станку вызываем для него `work_until`, проверяем, что запрос не игнорируется, добавляем задачу в очередь и увеличиваем `machine.finish` на ее длительность. При отмене задачи просто помечаем задачу отмененной, выставляем ей конец обработки и, возможно, начало, после чего уменьшаем `machine.finish` для соответствующего станка на `job.remaining()`.

В итоге получаем решение, работающее за время  $\mathcal{O}(n + q)$ , хотя в некоторых тестах и решение за  $\mathcal{O}(nq)$  могло пройти по времени.

В этой задаче было два варианта, и они отличались определением времени начала, форматами ввода и вывода и небольшой заменой одного из запросов. Все детали реализации во втором варианте остаются теми же, только, из основных изменений, время начала обработки задачи следует выставлять в другом месте: при добавлении задачи, если очередь пустая или задача срочная, а также при завершении обработки предыдущей в очереди задачи. Для этого удобно в `work_until` заменить `while machine.t < time` на `while machine.t <= time`.

## Бинарная Сила

В этой задаче можно было попробовать использовать жадные идеи — стараться сначала выполнять более дорогие квесты, но они не дали бы оптимальный ответ в большей части случаев. Так что будем оптимизировать полный перебор всех вариантов с помощью динамического программирования.

Динамическое программирование позволяет нам заметить, что если бы мы делали полный перебор, то при выборе навыка в очередной вершине нам были бы важны только значения навыков в ее детях, но не ниже. Соответственно, достаточно не перебирать все возможные распределения навыков по дереву, а только для каждой пары из вершины и ее возможного навыка определить оптимальный ответ на ее поддереве. Введем  $\text{dp}[v][i]$  — оптимальный ответ на поддереве  $v$ , если в ней стоит навык  $i$ .

Теперь надо научиться пересчитывать эти значения. Порядок пересчета понятен — просто считаем снизу вверх по дереву, тогда при вычислении  $\text{dp}[v]$  значения  $\text{dp}$  для ее детей уже посчитаны. Теперь заметим, что случаи, когда  $i$  находится в левом поддереве  $v$ , и в правом, симметричны, поэтому далее мы разберем только первый случай, а второй будет аналогичен.

Для удобства реализации пронумеруем все вершины сверху вниз и слева направо, начиная с нуля. Тогда несложно заметить, что дети вершины  $v$  будут иметь номера  $2v + 1$  и  $2v + 2$ . При этом номер слоя вершины  $v$  будет равен  $\lfloor \log_2(v + 1) \rfloor$ , что мы обозначим за  $\text{layer}(v)$ , а номера листьев в ее поддереве (в нумерации с нуля) будут лежать в интервале от  $\min(v) = (v + 1) \cdot 2^{n - \text{layer}(v)} - 2^n$  до  $\max(v) = \min(v) + 2^{n - \text{layer}(v)}$  не включительно. Скажем, что  $\text{dp}[v][i]$  отвечает за лист номер  $\min(v) + i$ .

При этом также сразу оценим, что сумма количества листьев по всем поддеревьям дерева имеет размер порядка  $2^n \cdot n$ , так как каждый лист входит в  $n$  поддеревьев. Поэтому если мы будем пересчитывать  $\text{dp}[v][i]$  за  $O(1)$ , полная асимптотика времени решения будет равна  $O(2^n \cdot n)$ .

Пусть  $i \in [\min(v), \frac{\min(v) + \max(v)}{2})$ , то есть находится среди левой половины листьев в поддереве  $v$ . Тогда  $\text{dp}[v][i]$  подразумевает, что в вершине  $2v + 1$  (в левом ребенке) также был выбран навык номер  $i$ . При этом в правом ребенке может быть выбран любой навык, поэтому

$$\text{dp}[v][i] = \text{dp}[2v + 1][i] + \max_{j \in [\min(2v+2), \max(2v+2))} \text{dp}[2v + 2][j] + \text{REWARD}.$$

Здесь **REWARD** означает количество монет, которое мы получим за размещение  $i$ -го навыка в  $v$ -й вершине. Давайте считать, что все квесты типа «less» изначально выполнены (то есть в базе динамики положим значения, равные сумме монет за все квесты типа «less» с уровнем больше 0), тогда за такое действие мы получим:

- минус число монет за квесты «less» для  $i$ -го навыка и уровня, соответствующего  $v$  (действительно, до этого ограничение квеста выполнялось, а теперь перестало);
- монеты за квесты «exact» для  $i$ -го навыка и уровня, соответствующего  $v$ ;
- монеты за квесты «more» с тем же уровнем навыка по аналогичным причинам.

Чтобы быстро считать это число монет, заранее сгруппируем все квесты по навыкам, а в каждой группе — по уровню. И для каждого уровня сложим монеты из квестов «exact» и «more» и вычтем монеты за квесты «less». Это и будет значение **REWARD**, используемое выше.

Остается только заметить, что  $\max_j \text{dp}[v][j]$  тоже можно поддерживать во время пересчета  $\text{dp}$  в отдельном массиве  $\text{max\_dp}$ , добавив  $O(1)$  времени на каждый пересчет. Тогда в сумме весь пересчет отработает за время  $O(2^n \cdot n)$ , а ответ в конце будет находиться в  $\text{max\_dp}[0]$ .