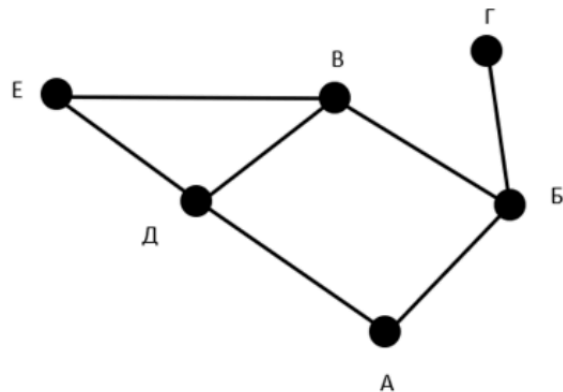


Номер 1

Задача на умение работать и графами и таблицей смежности(связей)

	П1	П2	П3	П4	П5	П6
П1		14		16	10	
П2	14		9			12
П3		9			13	
П4	16				8	
П5	10		13	8		
П6		12				



Давай посмотрим как работать с таблицей:

Тут самое главное умение - это умение играть в морской бой) Таблица указывает на то, есть ли путь между пунктами т е П1 связан с П2, П4, П5, смотрим на циферки или звездочки Для того чтобы узнать протяженность дороги, нужно просто посмотреть на пересечение пунктов, Например П1 - П2 это 14

Теперь давай посмотрим как работать с эскизом:

Ну думаю тут все предельно понятно, если есть линия между пунктами, то они связаны)) В принципе больше знать и не нужно

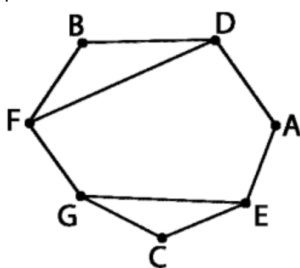
Для того чтобы его решить нужно лишь научиться сопоставлять таблицу и графы

Типы заданий

Симметричный граф(не явное задание графа)

На рисунке изображена схема дорог N-ского района. В таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет.

		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1		*	*	*			
	2	*		*			*	
	3	*	*					
	4	*				*		*
	5				*		*	*
	6		*			*		
	7				*	*		



Каждому населённому пункту на схеме соответствует номер в таблице, но неизвестно, какой именно номер. Определите, какие номера в таблице могут соответствовать населённым пунктам В и С на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

[Показать ответ](#)

37

Решение руками:

1. поиск индивидуальных путей по кол-ву связей
2. сопоставление по путям
3. нахождение того что просят В данном случае по таблице нет явных путей:

		Номер пункта							
		1	2	3	4	5	6	7	
Номер пункта	1		*	*	*				(3)
	2	*		*			*		(3)
	3	*	*						(2)
	4	*				*		*	(3)
	5				*		*	*	(3)
	6		*			*			(2)
	7				*	*			(2)

граф является симметричным, в таком случае нужно определять пункты по соседям. Заметим что только пункты F и G тройки связанные с (3), (3) и (2). Тогда 1 и 4 это F и G, следовательно 2ки, которые связаны с F и G являются искомыми В и С, значит В и С это 3 и 7. Мы явно не знаем какой пункт является каким, возможны 2 варианта:

1	2	3	4	5	6	7
F		B	G			C
G		C	F			B

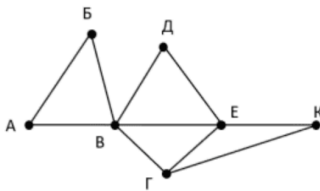
Но в данной задаче нас не спрашивают какой пункт под какой цифрой, так что и 3 7 и 7 3 нас полностью устроит, главное расположить все в том порядке в котором просят, в нашем случае по возрастанию, значит Ответ: 37

Несимметричный граф(явное задание графа)

№ 17772 (Уровень: Базовый)

(Д. Бахтиев) На рисунке схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7
п1		45		10			
п2	45			40		55	
п3					15	60	
п4	10	40				20	35
п5			15			55	
п6		55	60	20	55		45
п7				35		45	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите длину пути из пункта Д в пункт Е. В ответе запишите целое число.

[Показать ответ](#)

35

Решение руками:

1. поиск индивидуальных путей по кол-ву связей
2. сопоставление по путям
3. нахождение того что просят

В данной задаче мы можем явно определить 2 пункта

	П1	П2	П3	П4	П5	П6	П7	
П1		45		10				(2)
П2	45			40		55		(3)
П3					15	60		(2)
П4	10	40				20	35	(4)
П5			15			55		(2)
П6		55	60	20	55		45	(5)
П7				35		45		(2)

Тогда П4 - единственная четверка - Е

П6 - единственная пятерка - В

Заметим что Д это единственная 2ка, которая связана с В и Е

В(п6) –	2	3	4	5	7	
Е(п4) –	1	2			6	7
2 –	3ка					
7 –	2ка					

Тогда Д это П7 Так же можно определить пункт Г(единственная 3ка связанная с В и Е) Г это П2 Определить К будет довольно просто, единственный неизвестный пункт связанный с Е

Е(п4) –	1	2			6	7
	*	Г			В	Д

Тогда * это E

А и Б в данной задаче выяснить явно никак нельзя но для этого номера это и не требуется это либо А(п1) и Б(п5), либо А(п5) и Б(п1) Нужно найти путь ДЕ, следовательно Ответ: 35

Замечу, что немного неудачно подобрал задачи для этих 2х типов, цифры или звездочки могут быть и в 1м и во втором, типы влияют именно на симметричность, будет ли 1 вариант, как в нижнем или 2, как в верхнем

Решение кодом

Разница в 2х типах тут тоже наблюдается, но немного по-другому Решение будет полностью одинаково, как для симметричного так и для асимметричного, разница будет только в том как будет выводиться ответ

Шаблон кода:

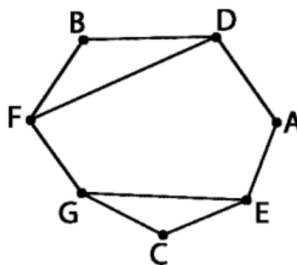
```
1 from itertools import permutations
2 a = 'связи через таблицу, какой с каким'.split()
3 s = 'ребра графа(дороги меж пунктов)'.split()
4
5 print('1 2...d'), где d – последняя цифра в таблице
6 for p in permutations('буквы графа')
7     if all(str(p.index(x) + 1) in a[p.index(y)] for x, y in s):
8         print(*p)
```

А теперь давай решим наши 2 задачи

№ 21887 Открытый вариант 2025 (Уровень: Базовый)

На рисунке изображена схема дорог N-ского района. В таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет.

		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1		*	*	*			
	2	*		*			*	
	3	*	*					
	4	*				*		*
	5				*		*	*
	6		*			*		
	7				*	*		



Каждому населённому пункту на схеме соответствует номер в таблице, но неизвестно, какой именно номер. Определите, какие номера в таблице могут соответствовать населённым пунктам В и С на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

[Показать ответ](#)

```

1  from itertools import permutations
2
3  a = '234 136 12 157 467 25 45'.split()
4  s = 'fb bd ba ae ec cg gf fd ge'.split()
5  print('1 2 3 4 5 6 7')
6  for p in permutations('abcdefg'):
7      if all(str(p.index(x) + 1) in a[p.index(y)] for x, y in s):
8          print(*p)

```

```

1 2 3 4 5 6 7
f d b g e a c
g e c f d a b

```

Можно взять любую из этих строк, при решении руками у нас тоже получилось 2 варианта

```

1 2 3 4 5 6 7
f d b g e a c – b(п3), c(п7)
Ответ: 37

```

```

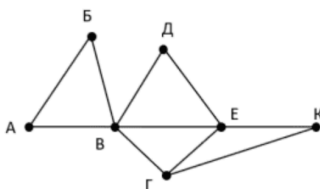
1 2 3 4 5 6 7
g e c f d a b – c(п3), b(п7)
Ответ: 37

```

№ 17772 (Уровень: Базовый)

(Д. Бахтиев) На рисунке схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о протяжённости каждой из этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7
п1		45		10			
п2	45			40		55	
п3					15	60	
п4	10	40				20	35
п5			15			55	
п6		55	60	20	55		45
п7				35	45		



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите длину пути из пункта Д в пункт Е. В ответе запишите целое число.

[Показать ответ](#)

35

```

1  from itertools import permutations
2
3  a = '24 146 56 1267 36 23457 46'.split()
4  s = 'аб ав вд ве вг де ге ек гк бв'.split()
5  print('1 2 3 4 5 6 7')
6  for p in permutations('абвгдек'):

```

```
7         if all(str(p.index(x) + 1) in a[p.index(y)] for x, y in s):
8             print(*p)
```

```
1 2 3 4 5 6 7
к г а е б в д
к г б е а в д
```

Так же берем любую(изменяются только а и б, что вообще не критично)

```
1 2 3 4 5 6 7
к г а е б в д – е(п4), д(п7), протяженность 35
Ответ: 35
```

```
1 2 3 4 5 6 7
к г б е а в д – е(п4), д(п7), протяженность 35
Ответ: 35
```

Объяснение кода

```
from itertools import permutations

a = '24 146 56 1267 36 23457 46'.split()
s = 'аб ав вд ве вг де ге ек гк бв'.split()
print('1 2 3 4 5 6 7')
for p in permutations('абвгдек'):
    if all(str(p.index(x) + 1) in a[p.index(y)] for x, y in s):
        print(*p)
```

```
from itertools import permutations – импортрование библиотеки для
permutations
```

```
a = '24 146 56 1267 36 23457 46'.split()
s = 'аб ав вд ве вг де ге ек гк бв'.split()
Переносим таблицу и граф в код, чтобы питон мог с ним работать
```

```
print('1 2 3 4 5 6 7') – выводим цифры в консоль для удобства
```

```
for p in permutations('абвгдек'):
```

перебираем все варианты а б в г д е к в различном порядке

Например:

а б в г д е к
а б в г д к е
а б в г е д к
а б в г е к д
а б в г к д е
...
к е д г в б а

Полностью все перестановки

Теперь единственная сложная строка, давай разберем максимально порционно и по порядку, но немного не так как написано Для генератора списка база, когда все работает с конца в начало

```
if all(str(p.index(x) + 1) in a[p.index(y)] for x, y in s):
```

Сначала я опущу все подробности, что как и почему оно работает, просто давай посмотрим что делает каждая часть этого if

1. `for x, y in s` перебирает различные варианты букв для x и y из s
'аб ав вд ве вг де ге ек гк бв'
сначала аб | x = а у = б потом ав | x = а у = в и так далее

2. `str(p.index(x) + 1) in a[p.index(y)]` смотрит на буквы из s
рассмотрим первую пару [аб] из s

`str(p.index(x) + 1)` становится цифрой 1й буквы
`in a[p.index(y)]` проверяет есть ли этот символ в "а" под номером 2й буквы

т е код проверяет есть ли связь в таблице между а и б

3. `all()` следит за тем, чтобы для всех пар из
s = 'fb bd ba ae ec cg gf fd ge'
есть связь в таблице
a = '234 136 12 157 467 25 45'


```
if all(str(p.index(x) + 1) in a[p.index(y)] for x, y in s):
```

проверяет что все что есть в "s" есть и в "a"

```
for p in permutations('abcdefg'):
```

```
    if all(str(p.index(x) + 1) in a[p.index(y)] for x, y in s):
```

проверяет что все что есть в "s" есть и в "a"

не для одной вариации, а для всех возможных

сначала для

а б в г д е к

потом для

а б в г д к е

и так далее

И как только `all` скажет, что такая вариация нам подходит, то надо ее вывести в консоль

```
print(*p) - выводит набор (а б в г д е к) на консоль
```

Полное объяснение что как и почему:

1. `for x, y in s` - перебирает символы из `s` для `x` и `y`

'[аб] ав вд ве вг де ге ек гк бв'

т е: `x = a y = б`

после

'аб [ав] вд ве вг де ге ек гк бв'

т е: `x = а у = в`

и так до конца

Итак, самое главное `for x, y in s` подставляет буквы из `s` под переменные `x` и `y`

2. `p.index(x) + 1 in a[p.index(y)]`

Разберем сейчас 1 шаг алгоритма:

Вспомним, что под `x` и `y` стоят разные буквы, например сейчас `x = а у = б`

т е номинально на данном конкретном этапе

`p.index('а') + 1 in a[p.index('б')]`

`p.index('что-либо')` даст нам номер позиции из строки `p`

на данном этапе `p` это а б в г д е к

`p.index('а') = 0`, так как `а` стоит на 0м месте

+ 1 нужен для того чтобы начинать не с 0, а с 1
`str(p.index('a') + 1)` просто преобразует 1 в '1'

`p.index('6') = 1`, так как 6 стоит на 1м месте
1 мы не добавляем так как будем смотреть в массиве "a" на
соответственной позиции, там тоже все начинается с 0

дальше мы смотрим содержится ли "1" а[1]
те '1' в 146

И так для каждой буквы на каждом шаге

3. теперь немного про `all()`

когда мы ищем наши пары с помощью `str(p.index(x) + 1) in a[p.index(y)]`
`for x, y in s` (далее буду называть A)

A – это генератор списков, т е после всего он выдаст список для каждой пары

если например а-б есть в таблице, то это будет 1, если же нет, то 0

после того как циклы пройдутся по всем парам у нас будет список

например [1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

`all` просто объединит все.

Если все в списке 1, то он выдаст `True` и наш `if` сможет для нас вывести p
Если же есть хотяб один 0, то он выдаст `False` и `if` не сработает, после
чего мы перейдем к другому набору символов из `permutations`