

Задача заключается в резервировании товаров из заказа со складов. Человек сделал заказ, нужно определить с каких складов осуществлять отправку.

Задача содержит в себе элементы оптимизации и комбинаторики, в частности "Задача об упаковке в контейнеры" и "Задача раскроя".

Входные данные

Warehouses - это словарь всех складов, ключ это идентификатор склада, а значение это приоритет этого склада в виде кортежа:

```
{warehouse_id: (int, int), ...}
```

Например:

```
warehouses = {1: (1, 0), 2: (2, 0) 3: (3, 0)}
```

Кортеж используется для удобства, чтоб в дальнейшем можно было добавить еще одно значение, для обозначения большего приоритета среди равных.

Приоритет у разных складов может быть одинаковым. Чем меньше значение тем выше приоритет.

Products - это все товары которые находятся на складах. Представляет собой словарь вида:

```
{product_id: {warehouse_id: quantity}, ...}
```

Например:

```
products = {1: {2: 7, 4: 3}, 13: {2: 7, 4: 3}, 21: {2: None, 4: None}}
```

Options - это все опции которые хранятся на складах, независимо от товаров. Представляет собой словарь вида:

```
{option_id: {warehouse_id: quantity}, ...}
```

Например:

```
options = {3: {2: None, 4: None}, 10: {2: 10}, 9: {2: 10, 4: 10}, 8: {4: 2}}
```

quantity - это количество на складе данного товара/опции. Если указано None значит товар/опция бесконечный. Если товар/опция бесконечный значит он на всех складах будет таким т.е.

невозможна ситуация {1: None, 2: 10}, всегда будет {1: None, 2: None}

Order - это заказ(корзина) в котором лежат товары и выбранные к ним опции. Один и тот же товар с разными комбинациями опций рассматривается как отдельный. Кол-во в данном случае определяет кол-во данного товара и кол-во каждой опции в данной комплектации, т.е. при количестве 2шт. нужно взять 2шт. товара и по 2шт. каждой опции.

Имеет следующий вид:

```
[{'product_id': product_id, 'quantity': int, 'options': [option_id, ...]}, ...]
```

Например:

```
order = [  
  {'product_id': 1, 'quantity': 2, 'options': [3, 10, 9]},  
  {'product_id': 1, 'quantity': 1, 'options': [3, 9]},  
  {'product_id': 13, 'quantity': 5, 'options': [10, 9]}  
]
```

В данном случае человек заказал три товара, первые два это один и тот же товар, но т.к. у него разные комбинации опций он идет как два разных товара, по сути это как два вида/позиции одного и того же товара.

Должны быть реализована функция в которую передается четыре аргумента: warehouses, products, options, order.

На выходе мы должны получить кортеж (orders, limit_exceeded).

Выходные данные

Orders - это массив вида:

```
[{'warehouse': warehouse_id, 'order_products': [order_product_index, ...]}, ...]
```

order_products - это массив с индексами из массива Order

Например:

```
orders = [  
  {'warehouse': 1, 'order_products': [0, 1, 2]},  
  {'warehouse': 2, 'order_products': [3]},  
]
```

По сути мы берем массив Order и разбиваем его на заказы(Orders), дабы сгруппировать заказанные товары с опциями по складам, для того чтобы знать откуда, что и сколько отправлять.

Limit exceeded - это массив вида:

```
[{'order_product': order_product_index, 'max_quantity': int}, ...]
```

В нем находятся товары из заказа у которых превышено максимальное кол-во которое доступного для заказа.

max_quantity - должно отражать максимально доступное кол-во для заказа, оно нужно чтобы потом отобразить его покупателю.

Подсчет этого значения нужно производить в самом конце, когда все товары которые возможны для заказа были зарезервированы(добавлены в orders) по складам.

Примеры

Пример 1.

```
warehouses = {1: (4, 0), 2: (3, 0), 3: (2, 0), 4: (1, 0)}
products = {1: {1: 7, 2: 3}, 2: {2: 7, 3: 3}}
options = {9: {2: 10, 3: 2, 4: 10}}
order = [{product_id: 1, 'quantity': 1, 'options': []}, {product_id: 2, 'quantity': 3, 'options': [9]}
```

Результат:

```
orders, limit_exceeded = ({'warehouse': 2, 'order_products': [0, 1]}, [])
```

order[0] - резервируем со 2ого склада т.к. его приоритет выше чем 1ого, а товар есть только на двух этих складах.

order[1] - товар есть на складе 2 и 3, а опция связанная с ним на складе 2, 3, 4.

Получается, что товар мы могли бы отправить только со складов 2 и 3, руководствуясь приоритетом складов мы бы выбрали склад 3.

Но на 3ем складе нет нужного кол-во опций, т.к. заказали 3шт. и нужное кол-во товара и опции есть только на складе 2, поэтому резервируем его со склада 2.

limit_exceeded - пустой, т.к. все товары можно отправить.

Пример 2.

```
warehouses = {1: (4, 0), 2: (3, 0), 3: (2, 0), 4: (1, 0)}
products = {1: {2: None, 4: None}, 13: {2: 40, 4: 50}}
options = {3: {2: 20, 4: 30}, 9: {2: None, 4: None}}
order = [
    {product_id: 1, 'quantity': 40, 'options': [3, 9]},
    {product_id: 13, 'quantity': 5, 'options': [3, 9]}
]
```

Результат:

```
orders, limit_exceeded = (
    [{'warehouse': , 'order_products': [1]}],
    [{'order_product': 0, 'max_quantity': 25}]
)
```

order[0] - сам товар бесконечен, но опция 3 связанная с ним нет и ее не хватает ни на одном складе, поэтому заносим данный товар в limit_exceeded.

Значение max_quantity равно 25 т.к. после того как мы зарезервируем следующий товар, который ссылается на 3 опцию, на 4ом складе этой опции останется 25.

order[1] - отправим со склада 4.

Пример 3.

```
warehouses = {1: (1, 0), 2: (3, 0), 3: (2, 0), 4: (1, 0), 12: (10, 0), 10: (12, 0)}
```

```
products = {
```

```
    1: {1: None, 2: None, 3: None, 12: None},
```

```
    2: {1: None, 2: None, 3: None, 12: None},
```

```
    3: {1: None, 2: None, 3: None, 12: None},
```

```
    4: {1: None, 2: None, 3: None, 12: None},
```

```
}
```

```
options = {3: {1: 50, 2: 50, 3: 50, 10: 200}, 4: {3: 20}}
```

```
order = [
```

```
    {'product_id': 1, 'quantity': 10, 'options': [3]},
```

```
    {'product_id': 2, 'quantity': 10, 'options': [3, 4]},
```

```
    {'product_id': 3, 'quantity': 10, 'options': [3, 4]},
```

```
    {'product_id': 4, 'quantity': 10, 'options': [3]},
```

```
]
```

Результат:

```
orders, limit_exceeded = (
```

```
    [
```

```
        {'warehouse': 1, 'order_products': [0]},
```

```
        {'warehouse': 3, 'order_products': [1, 2, 3]}
```

```
    ],
```

```
    []
```

```
)
```

order[3] - отправим со склада 3. т.к. с него можно отправить большее кол-во товаров.

Если приоритет одного склада выше приоритета другого склада, но с другого склада можно отправить больше товаров, то нужно выбрать другой.

"Больше товаров" означает именно кол-во видов/позиций, а не quantity.

Полный список приоритетов выбора склада написаны внизу.

Уточнения:

Один заказанный товар разбивать нельзя, т.е. если заказали 5 шт. и ни на одном складе нет такого кол-ва, то его заносим в limit_exceeded, даже если его можно отправить по 1шт с пяти складов.

Нужно стараться отправить все товары с минимального кол-ва складов, учитывая все условия/приоритеты выбора склада.

Приоритет заказанных товаров между друг другом определяется их порядком в массиве.

Если товар возможно отправить с нескольких складов, то порядок определения склада по убыванию:

1) Наибольшее кол-во товаров которое возможно зарезервировать с одного склада.

2) Приоритет склада в словаре warehouses.

3) Наибольшее кол-во штук всех товаров отправляемое со склада, именно штук. Кол-во видов/позиций товара у нас на первом месте.

4) Минимальная позиция товара(ключ из order) который уже зарезервирован на складе.

5) Склад на котором меньше всего данного товара.

6) Если все вышеописанные условия равны, то выбираем рандомно любой из возможных складов.