Полезные приёмы для решения заданий 26 ЕГЭ по информатике на Python

Пилосов Петр Анатольевич
учитель информатики МАОУ СОШ №200

Чем интересно 26 задание?

- Задачи имеют прикладной характер
- Задачи различны по типу и сложности
- Задачи требуют внимательного, вдумчивого прочтения теста, понимания, что от тебя хотят
- Задачи не так сложны технически, как это кажется с первого раза
- Задачи имеют множество способов решения
- Задачи могут решаться не только методами программирования*
- Ответ можно найти, не оформляя полное решение*

** во многих случаях*

Технические идеи для решения задач 26 на Python

- Научиться работать со вложенными списками. Вложенные списки в Python — это списки, элементами которых являются другие списки
- Научиться методам сортировки с помощью лямбда-функций
- «Перебрасывать» данные из одного списка в другой при необходимости, делая срезы, меняя условия и пр.
- «Подглядывать» промежуточные ответы, используя их для нахождения окончательного
- Использовать ключи (при наличии) для самопроверки

26_1

Отбор кандидатов в матросы происходит по сумме баллов трех экзаменов. На заранее известное количество мест отбираются кандидаты, набравшие большую сумму баллов по результатам трех экзаменов. **Все кандидаты, набравшие определенную сумму баллов или больше, зачисляются на имеющиеся места. Такой балл называется проходным.** Если после заполнения имеющихся мест кандидатами с проходным баллом остаются незаполненные места, но кандидатов, набравших следующую сумму баллов, больше чем вакантных мест, набранная этими кандидатами сумма баллов называется полупроходным баллом. **Из числа кандидатов, набравших полупроходной балл, на имеющиеся места принимаются кандидаты, имеющие более высокий балл за собеседование, а при равенстве баллов за собеседование – приоритет имеют кандидаты с наименьшими ID.**
Для данного множества кандидатов следует определить ID последнего кандидата с набранным проходным баллом, а также каково количество кандидатов, набравших полупроходной балл.

Входные данные
В первой строке входного файла находится два числа N – количество кандидатов (натуральное число, не превышающее 1000) и S – количество имеющихся мест. Каждая из следующих N строк содержит пять чисел: ID кандидата (натуральное число, не превышающее 10 000), соответственно три оценки по экзаменам (все числа целые неотрицательные, не превышающие 100) и балл за собеседование (целое неотрицательное число, не превышающее 10).
Запишите в ответе два целых числа: сначала ID последнего кандидата с набранным проходным баллом, а затем количество кандидатов, набравшие полупроходной балл.

Ответы к заданию:

7600410 14

Типовой пример организации данных во входном файле
6 3
1 90 90 90 10
3 60 70 80 8
5 63 60 90 6
8 50 80 100 4
4 40 95 80 7
11 80 63 72 6
При таких входных данных проходной балл равен 230, полупроходной 215, на оставшееся одно место будет назначен кандидат, набравший в сумме 215 баллов и получивший по собеседованию 7 баллов.

Ответ для приведённого примера: 8 2.

Сортировка по лямбда-функции

```
users.sort(key=lambda x:(-x[1],-x[2],x[0]))
```

Что здесь происходит?

Главный параметр сортировки – по убыванию

Вспомогательный (если x[1] совпал) по убыванию

Дополнительный (если x[1] и x[2] совпали) – по возрастанию

Обязательны скобки!!!

```
f=open('26_1.txt')
n,k=map(int,f.readline().split())
#n-число абитуриентов, k-число вакансий
users=[]
#список данных - id, баллы, собеседование

for i in range(n):
    idp,b1,b2,b3,sb=map(int,f.readline().split())
    s=b1+b2+b3
    users+=[[idp,s,sb]]
#сортируем по сумме и во вторую очередь по собеседованию, в третью - по id
users.sort(key=lambda x:(-x[1],-x[2],x[0]))
z=users[:k]
print(z[-10:])
#в этот момент достаточно подглядеть проходной балл (155) и полупроходной(154)
pp=0
for x in users:
    if x[1]==154:
        pp+=1
print(pp)
#id проходного балла подсмотрите самостоятельно!
```

26_2

В камере хранения аэропорта есть *K* ячеек для хранения багажа туристов. Все ячейки пронумерованы, начиная с единицы. Известно время, в которое каждый турист придёт оставить свой багаж, и в какое время он заберёт его. С приходом каждого туриста его багаж кладётся в свободную ячейку с наименьшим номером. Для того, чтобы разгрузить или загрузить ячейку багажом, необходима 1 минута. Со следующей минуты можно положить в освободившуюся ячейку багаж другого туриста. Если турист пришёл, но свободных ячеек нет – он багаж оставить не может, поэтому уходит.

Определите, сколько всего туристов придут и оставят свой багаж в ячейках за 24 часа и номер ячейки, в которую положат последний багаж. Если вариантов выбрать ячейку несколько – выберите свободную ячейку с наименьшим номером.

Входные данные

В первой строке входного файла находится число *K* – количество ячеек в аэропорту (натуральное число, не превышающее 1000). Во второй строке находится число *N* – количество туристов, которые собираются воспользоваться ячейками для багажа. В следующих *N* строках находятся два значения: минута размещения багажа и минута, до которого планируется хранить багаж в ячейке, отсчёт ведётся от начала суток (все числа неотрицательные, не превышающие 1440), для каждого туриста – в отдельной строке.

Запишите в ответе два целых числа: сначала количество туристов, которое сможет воспользоваться ячейками для багажа за 24 часа, затем наименьший номер ячейки, в которую положат последний багаж.

Ключ для обсуждения с учениками в классе

2	8			itog[j]			
9	13		1 яч	0	[2,8]	[9,13]	[14,45]
10	27		2 яч	0	[10,27]		
12	17		3 яч	0	[12,17]	[18,20]	[21,24]
14	45						
15	20		Можно хранить только время, когда забрали багаж				
16	18						
18	20						
21	24		Придумайте свой способ решения *				
23	27						

Ответы к заданиям:

586 3

```
f = open('26_2.txt')
k = int(f.readline())  #ячеек багажа
n = int(f.readline()) #пассажиров
b = []
for i in range(n):
    t1, t2 = map(int, f.readline().split())
    b.append([t1, t2])
#сортируем багаж по времени прибытия пассажира
b.sort()
print(b[:10])
f = 0
# имеет смысл хранить только время, до которого ячейка занята
itog = [0] * k
#идёт пассажир
for i in range(n):
    #предлагаем ему k ячеек
    for j in range(k):
        #если время свободно, занимаем и обновляем список itog
        if b[i][0] > itog[j]:
            itog[j] = b[i][1]
            #считаем тех, кто положил, запоминаем время когда j ячейка освободится
            f += 1
            r = j + 1
            # и сразу переходим к новому пассажиру!
            break
print(f, r)
```

26_3

Отдел маркетинга сети магазинов составляет рейтинг продуктов по информации об их сроках хранения с момента изготовления и после вскрытия упаковки. Для каждого продукта известен срок его хранения с момента изготовления и срок годности к употреблению после вскрытия упаковки. Продукты **пронумерованы начиная с единицы**.

В рейтинговом списке маркетологи располагают продукты по следующему алгоритму:
– **все 2N чисел**, обозначающих срок хранения и срок годности к употреблению для N продуктов, **упорядочивают по возрастанию;**
– если минимальное число в этом упорядоченном списке – срок хранения, то продукт в рейтинге занимает **первое свободное место от его начала;**
– если минимальное число – срок годности к употреблению, то продукт занимает **первое свободное место от конца рейтинга**;
– если число обозначает срок хранения или срок годности к употреблению уже рассмотренного продукта, то его не принимают во внимание.

Этот алгоритм применяется последовательно для размещения всех N продуктов.

Определите номер последнего продукта, для которого будет определено его место в рейтинге, и количество продуктов, которые займут в рейтинге более низкие места.

Входные данные
В первой строке входного файла находится натуральное число N (N ≤ 1000) – количество продуктов. Следующие N строк содержат пары чисел, обозначающих соответственно срок хранения продукта с момента изготовления и срок годности к употреблению после вскрытия упаковки (все числа натуральные, различные).

Запишите в ответе два натуральных числа: сначала **номер последнего продукта, для которого будет определено его место** в рейтинге, затем – **количество продуктов, которые займут в рейтинге более низкие места.**

Ключ

Типовой пример организации данных во входном файле

5
30 50
100 155
150 170
10 160
120 55

При таких исходных данных порядок расположения продуктов в рейтинге следующий: 4, 1, 2, 3, 5. Последним займёт своё место в рейтинге продукт 3. При этом один продукт займёт в рейтинге более низкое место.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемых файлов.

Ответы к заданию

564 444

```
f=open('26_3.txt')
n=int(f.readline())
goods=[]
k=0
#t1-срок хранения (0), t2-срок годности после вскрытия (1)
for line in f:
    k+=1
    t1,t2=map(int,line.split())
    if t1<t2:
        goods.append([t1,k,0])
    else:
        goods.append([t2, k, 1])

#сортируем товары по времени
goods.sort()
#печатаем последний товар, его номер – 564 (первый ответ)
print(goods[-1])
final=[]
#дальше можно составить рейтинг, а можно просто догадаться
# т.к. последний товар имеет маркер "1", то все товары ниже него - все единицы (кроме него!)
t=0
for x in goods:
    if x[2]==1:
        t+=1
print(t-1)

#Ответ 2 – 444.
Альтернативная концовка – сделать рейтинговый список строго по условиям задачи (дольше)
# for x in goods[::-1]:
#     if x[2]==0:
#         final=[x]+final
#     else:
#         final = final+[x]
# t=0
# for x in final:
#     t += 1
#     if x[1]==564:
#         break
# print(n-t)
```

26_4

№ 23570 Пересдача 03.07.25 (Уровень: Сложный)

Для дачных участков СНТ необходимо закупить снегоуборщики. Для каждого из N участков будет куплен свой снегоуборщик. Известны минимальные требования к мощности этой техники для каждого из участков.

Для закупки доступно K моделей снегоуборщиков определённой мощности и стоимости. Количество экземпляров каждой модели не ограничено. Для каждого участка выбирается снегоуборщик минимальной стоимости, мощность которого не меньше требуемой; при одной и той же стоимости выбирается модель максимальной мощности.

Требуется определить общую стоимость закупки и максимальную мощность снегоуборщика, входящего в число купленных. В ответе запишите два числа: сначала суммарную стоимость всех купленных снегоуборщиков, затем максимальную мощность среди них.

Входные данные

Первая строка входного файла содержит два натуральных числа: N (1 < N < 1 000 000) - количество участков СНТ и K (1 < K < 100 000) - количество моделей снегоуборщиков соответственно.

Следующие N строк содержат по одному натуральному числу, не превышающему 1000, минимальные мощности снегоуборщиков, которые можно закупить для каждого из N участков. Далее в каждой из K строк содержится пара натуральных чисел - мощность очередной модели снегоуборщика и её стоимость соответственно. Мощность снегоуборщиков не превосходит 1000, стоимость - 100 000. Гарантируется, что любые две модели снегоуборщиков различаются по мощности или по стоимости. Закупить подходящий набор снегоуборщиков всегда можно.

Выходные данные

В ответе укажите два искомых числа: суммарную стоимость всех купленных снегоуборщиков и максимальную мощность среди них.

Типовой пример организации данных во входном файле

3 4
1
2
3
10 7
1 5
3 7
2 3

При таких исходных данных для первого и второго участков оптимально закупить одинаковые снегоуборщики мощностью 2 и стоимостью 3, для третьего участка будет закуплен снегоуборщик мощностью 10. Стоимость закупки составит 3 + 3 + 7 = 13. Ответ: 13; 10.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемого файла.

1 часть задачи

```
f=open('26_4.txt')
n,k=map(int,f.readline().split())
m=[]
#считываем минимальную мощность для каждого участка
for i in range(n):
    m+=[int(f.readline())]
s=[]
#считываем цену pr снегоуборочной машины и её мощность pw
for j in range(k):
    pw,pr=map(int,f.readline().split())
    s+=[[pr,pw]]
print(len(s),len(m))
s.sort(key=lambda x:(x[1],x[0]))
print(s[:10])
```

Проблема – почти 100000 машин на 200000 участков, программа не проходит по скорости. Что делать? Оптимизировать список s – для машины каждой мощности (а их всего 1000) оставить наименьшую цену.

2 часть задачи

```
#Оптимизация: оставляем для каждой мощности минимальную цену (список s1)
s1=[s[0]]
for i in range(k-1):
    if s[i][1]<s[i+1][1]:
        s1.append(s[i+1])

#сортируем по цене (возрастание) и мощности(убывание)
s1.sort(key=lambda x:(x[0],-x[1]))
final=[]
#подбираем для каждого участка агрегат согласно условиям
#благодаря сортировке это просто – минимальная цена и максимальная мощность
for i in range(n):
    for j in range(len(s1)):
        if s1[j][1]>=m[i]:
            final+=[s1[j]]
            break

#находим общую стоимость и максимальную мощность
h,m=0,0
for x in final:
    h+=x[0]
    m=max(m,x[1])
print(h,m)
```

Рекомендации для подготовки

- Решать задания 26 двумя разными способами
- Поддерживать «авторские» решения учеников
- Учить «смысловому чтению» и проверке на файле ключа
- Оставить достаточно времени для решения (на экзамене)
- Рассказывать учащимся про скорость выполнения программ (включать библиотеку time)
- Учить печатать и «подсматривать» ответы – включая промежуточные

Спасибо за внимание

Коды программ можно копировать в любой Питон. Сами файлы заданий вы можете легко найти и скачать на сайте

https://kompege.ru/