

Вступление

Всем привет! Сегодня я буду проходить вместе с вами тестирование для отбора кандидатов на **Стажировку от Яндекс**. И, хоть мы в последнее время и изучаем **Data Engineering**, попасть на работу сейчас будет очень полезно. Поэтому давайте начнем, у нас всего лишь **6 часов**.

Попробовать свои силы, и, возможность, стать сотрудников Яднекса можно по прикрепленной ссылке ниже:

[Нажми Нажми Нажми Нажми Нажми Нажми Нажми Нажми Нажми](#)



Приступаем к решению задач!

Очевидное

Конечно же, я нашел эти задачи в интернете, и не ручаюсь за их достоверность. Весь приведенный материал показывается исключительно в ознакомительных целях, все совпадения случайны, и нигде вообще не написано, что **даже** если вдруг **это я** реально решал эти задачи сам **6 часов** 2023.09.25, то мне что-то за это будет, **отстаньте**.

Задача 1

А. Лента событий SQL

- ✓ А. Лента событий SQL
- ✓ В. Код ревью
- С. Даркстор
- ✓ D. Случай в казино
- ✓ E. Игра с шариками
- Ф. Дикие неончики

Ограничение времени	10 секунд
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Задание на знание языка SQL.

Дана таблица с логом показов рекламы в ленте:

```
feed_events (  
  uuid: str,  
  timestamp: int,  
  event: str  
)
```

, где `uuid` — `id` пользователя, `timestamp` — Unix-время события, `event` — тип события. В поле `event` может быть одно из трех значений: `'open'` — открытие ленты, `'show'` — показ рекламы, `'click'` — клик в рекламу.

Выведите 10 `uuid`'ов, у которых было наибольшее число сессий с хотя бы одним кликом в рекламу, с самым числом сессий. Началом сессии будем считать событие `'open'`, концом сессии — следующее событие `'open'` у этого `uuid`'а (это же событие начинает новую сессию). Результат должен быть отсортирован в порядке убывания числа сессий. Если у разных `uuid`'ов число сессий совпадает, упорядочьте их лексикографически.

Запрос должен быть написан на диалекте SQLite3. Обратите внимание, что в SQLite может не быть привычных вам по другим СУБД функций, сверяйтесь с [документацией](#). Также вместо создания таблиц в базе данных стоит обходиться подзапросами и CTE.

Формат ввода

uuid	timestamp	event
9729e...	1639667950	open
9729e...	1639716580	show
f2a5b...	1639754451	open
9729e...	1639802019	click
9729e...	1639829648	open
2ef5c...	1639849650	open
9729e...	1639879652	show
9729e...	1639889750	click
9729e...	1639909751	open
2ef5c...	1639929851	show
2ef5c...	1639979950	click
2ef5c...	1639999999	open

Формат вывода

uuid	sessions
9729e...	2
2ef5c...	1
f2a5b...	0

Примечания

Для упрощения задачи гарантируется, что первое и последнее событие для каждого `uuid`'а — событие `'open'`.

Примечание

Думаю, не нужно объяснять код, потому что он максимально структурирован, хоть и не идеален. Я старался максимально разделить код на отдельные микроструктуры.

```
with
t1 AS(
  SELECT *, row_number() over (partition by uuid) full_sessions
  from feed_events
  WHERE event = 'open'),
t2 AS(
  SELECT feed_events.*, t1.full_sessions
  FROM feed_events
  LEFT JOIN t1
  ON t1.uuid=feed_events.uuid
  AND t1.timestamp=feed_events.timestamp),
t3 AS(
  SELECT *
  FROM t2
  ORDER BY uuid asc,timestamp asc),
t4 AS(
  SELECT *,
  coalesce(full_sessions,
    (SELECT max(t3.full_sessions)
     from t3
     where t3.uuid=t3_1.uuid
     AND t3.timestamp<t3_1.timestamp)
  )) sessions_num
  from t3 t3_1),
t5 as(
  SELECT DISTINCT uuid,sessions_num
  from t4
  where event='click'),
t6 AS(
  SELECT uuid, count(sessions_num) sessions
  from t5
  group by uuid),
t7 as(
  select distinct uuid, min(event) gh
  from t4
  group by uuid
  having min(event)<>'click'
),
t8 as(
  select * from t6
  UNION
  SELECT uuid, 0 sessions
  from t7)
SELECT * from t8
order by sessions desc,uuid asc
LIMIT 10
```

Задача 2

В. Код ревью

Ограничение времени	20 секунд
Ограничение памяти	2.0 Гб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

- ☒ A. Лента событий SQL
- ☒ B. Код ревью
- ☐ C. Даркстор
- ☒ D. Случай в казино
- ☒ E. Игра с шариками
- ☐ F. Дикие неончики

В Яндексе был стажер по имени Ян, замечательный друг и хороший человек. Недавно он получил задачку и буквально сегодня закончил её выполнение. Дело оставалось за малым: его код должен пройти ревью. Для этого необходимо, чтобы с ним ознакомились зависимые от предлагаемых Яном изменений команды.

Вот только проблема: некоторые команды не хотели тратить время на проверку кода, ссылалась на большую загруженность, и предлагали направить код на проверку другим командам. Ян решил подойти к ревью очень серьезно и начал опрашивать команды для составления списка всех возможных перенаправлений. В результате получился довольно внушительный документ.

Ян начал чувствовать себя как в замкнутом круге. Он бегал от команды к команде, каждая ссылалась на другую. В какой-то момент он начал думать, что они могут ссылаться друг на друга и в таком случае нет никакого шанса на то, что его изменения увидят мир!

Вы решили не оставаться в стороне и помочь Яну (он же замечательный друг и хороший человек). Вы сказали, что сможете проанализировать полученный документ и ответить на вопрос, есть ли в нем циклические зависимости.

Формат ввода

Первая строка содержит целое число N — количество элементов в списке.

Вторая строка содержит N целых чисел, разделенных пробелами, принимающие значения в диапазоне от -1 до $N - 1$ включительно.

Значение под номером i говорит о том, на какую команду ссылается команда под номером i . Нумерация команд начинается с нуля. Значение -1 говорит об отсутствии ссылки на какую-либо команду.

- $1 \leq N \leq 10^7$;
- $-1 \leq A[i] \leq N - 1$, где A — список чисел, описывающих перенаправления команд;
- $A[i] \neq i$.

Формат вывода

Выведите одно слово в зависимости от того, существуют ли циклические зависимости в требованиях команд.

YES — циклические зависимости существуют.

NO — в противном случае.

Пример 1

Ввод	Вывод
5 -1 -1 -1 -1 -1	NO

Пример 2

Ввод	Вывод
------	-------

5
1 3 -1 1 0

YES

Пример 3

Ввод



Вывод



5
-1 2 1 -1 -1

NO

Примечания

В первом примере команда Яна не перенаправляет код никакой другой команде.

Во втором примере команды 1 и 3 перенаправляют код друг другу.

В третьем хоть команды и ссылаются друг на друга, но так как команда Яна (нулевая) не ссылается ни на какую другую команду, то перенаправлений не будет.

Пояснение

На самом деле код оказался куда более простым, чем я изначально думал.

🧐 В чем же дело?

Всё дело в том, что **Яндекс** неправильно написал условие задачи.

Нас просят найти наличие циклических зависимостей **среди** всех команд. Однако на самом деле нужно найти эти зависимости только среди тех команд, которые перенаправили **Яна** из **его собственной** команды. Ниже код с учетом этого, и он оказался правильным. Дойти до этого можно, но это не очевидно, и выясняет только путем проверочных вброс-тестов.

```
N=input()
b=input()
b=[int(x) for x in b.split(' ')]
ender=False
ans='NO'
i=0
while ender==False:
    if b[i]==-1:
        ender=True
    if b[i]==-2:
        ans='YES'
        ender=True
    if b[i]!=-1 and b[i]!=-2:
        t=i
        i=b[i]
        b[t]=-2
    # print(f"{b}")

print(ans)
```

Задача 3

С. Даркстор

Ограничение времени	1 секунда
Ограничение памяти	512.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

- ✓ A. Лента событий SQL
- ✓ B. Код ревью
- | C. Даркстор
- ✓ D. Случай в казино
- ✓ E. Игра с шариками
- F. Дикие неончики

Петя стажёр в Яндекс Лавке и ему нужно решить задачу по размещению нового даркстора на карте размером N на M , где каждая клетка имеет определенное значение прибыли или убытка. Цель Пети - выбрать непустую прямоугольную область на карте, где разместить даркстор будет наиболее выгодно.

Формат ввода


Первая строка содержит два числа N и M - размеры карты. ($1 \leq N, M \leq 200$)


Следующие N строк содержат M чисел x_{ij} ($1 \leq i \leq N, 1 \leq j \leq M, -10^5 \leq x_{ij} \leq 10^5$) - значение прибыли или убытка для Яндекса в каждой клетке.

Формат вывода

Выведите одно число - максимальную возможную прибыль.

Пример 1


Ввод 


Вывод 

```
2 2
-2 1
2 4
```

6

Пример 2

Ввод 

Вывод 

```
3 3
1 0 4
2 -1 1
-1 5 2
```

13

Пояснения

Данная задача непомерно сложная по сравнению с другими, так как требует очень долгого анализа и кропотливой работы, потому что ограничение по времени и памяти слишком сильное.

Интересный факт

Данная задача использовалась на олимпиаде **Росатом** 2022-2023 годов. Найти ее можно в данном [файле](#), а именно задача №2. Решение там неполное, и очень сложное к пониманию (Это же не Python 😊)

Задача 4

D. Случай в казино

Ограничение времени	10 секунд
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

- ☒ A. Лента событий SQL
- ☒ B. Код ревью
- ☐ C. Даркстор
- ☒ D. Случай в казино
- ☒ E. Игра с шариками
- ☐ F. Дикие неончики

Аркадий пришел играть в необычное казино, где вместо привычных карт используется колода с напечатанными числами от 1 до N , а игра проводится так - крупье достает по одной карте из колоды и показывает число на ней; в любой момент, даже до первой показанной карты, Аркадий может сказать "стоп" и если число на следующей карте будет простым, то он побеждает, если нет - проигрывает. К удивлению Аркадия оказалось, что карты были заряжены в киоске (случайно перемешаны) и его знания математики не помогут выиграть гарантированно, однако они все равно помогли придумать стратегию, которая максимизирует вероятность выигрыша.

Найдите лучшую стратегию игры для N карт и в качестве ответа напишите вероятность выигрыша при игре с ней с точностью до второго знака после запятой.

Формат ввода


Одно целое число N , $17 \leq N \leq 10^7 + 9$.

Формат вывода

Одно число - вероятность победы при игре с лучшей стратегией, округленное до 2 знаков после запятой.

Пример

Ввод 

Вывод 

2

0.50

Пояснение

Очень интересная задача. Я, наверное, только час думал над тем, какой самый лучший метод нужно использовать для наибольшей вероятности выигрыша. И только спустя час, до меня дошло:

- **Стратегии нет!** От количества пропущенных карт, среднестатистическое отношение простых чисел к общему набору будет оставаться **всегда** прежним! А это в свою очередь значит, что **Аркадий** может останавливать игру в абсолютно любой момент, и в абсолютно любой момент его общая вероятность победы будет оставаться на прежнем уровне, а именно **(Кол-во Простых Чисел) / (Всего чисел)**. Всё, что нам остается сделать (иначе не пройдем по ограничениям), это использовать один из наиболее быстрых методов нахождения кол-ва простых чисел, это гуглится, есть быстрые алгоритмы. Ниже один из них.

```
n=int(input())
def primes(n):
    """ Returns a list of primes < n """
    sieve = [True] * n
    for i in range(3,int(n**0.5)+1,2):
        if sieve[i]:
            sieve[i*i::2*i]=[False]*((n-i*i-1)//(2*i)+1)
    return len([2] + [i for i in range(3,n,2) if sieve[i]])

a=primes(n)
```

```
t=round(float(a/n),2)
print(f'{t:.2f}')
```

Задача 5

Е. Игра с шариками

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

- ☒ А. Лента событий SQL
- ☒ В. Код ревью
- ☐ С. Даркстор
- ☒ D. Случай в казино
- ☒ Е. Игра с шариками
- ☐ F. Дикие неончики

Есть три мешочка: в первом 1 белый шар, 1 чёрный, во втором - 2 белых, 1 чёрный, в третьем - 6 белых и 1 чёрный. На каждом шаге из каждого мешочка вытаскивается по одному шару; если все три белых, то игра заканчивается; если хотя бы один из трёх чёрный, то вытасканные шары возвращаются обратно в мешочки, в каждый из них добавляется по одному чёрному шару, и указанный процесс повторяется (снова вытаскивается по одному шару из каждого мешочка и т.д.) Какова вероятность, что игра продлится не более n шагов?

Формат ввода

Одно целое число n , $1 \leq n \leq 10^9$.

Формат вывода

Два неотрицательных целых числа p и q таких, что p/q - несократимая дробь, равная вероятности, что игра продлится не более n шагов.

Пример 1

Ввод

Вывод

1

2 7

Пример 2

Ввод

Вывод

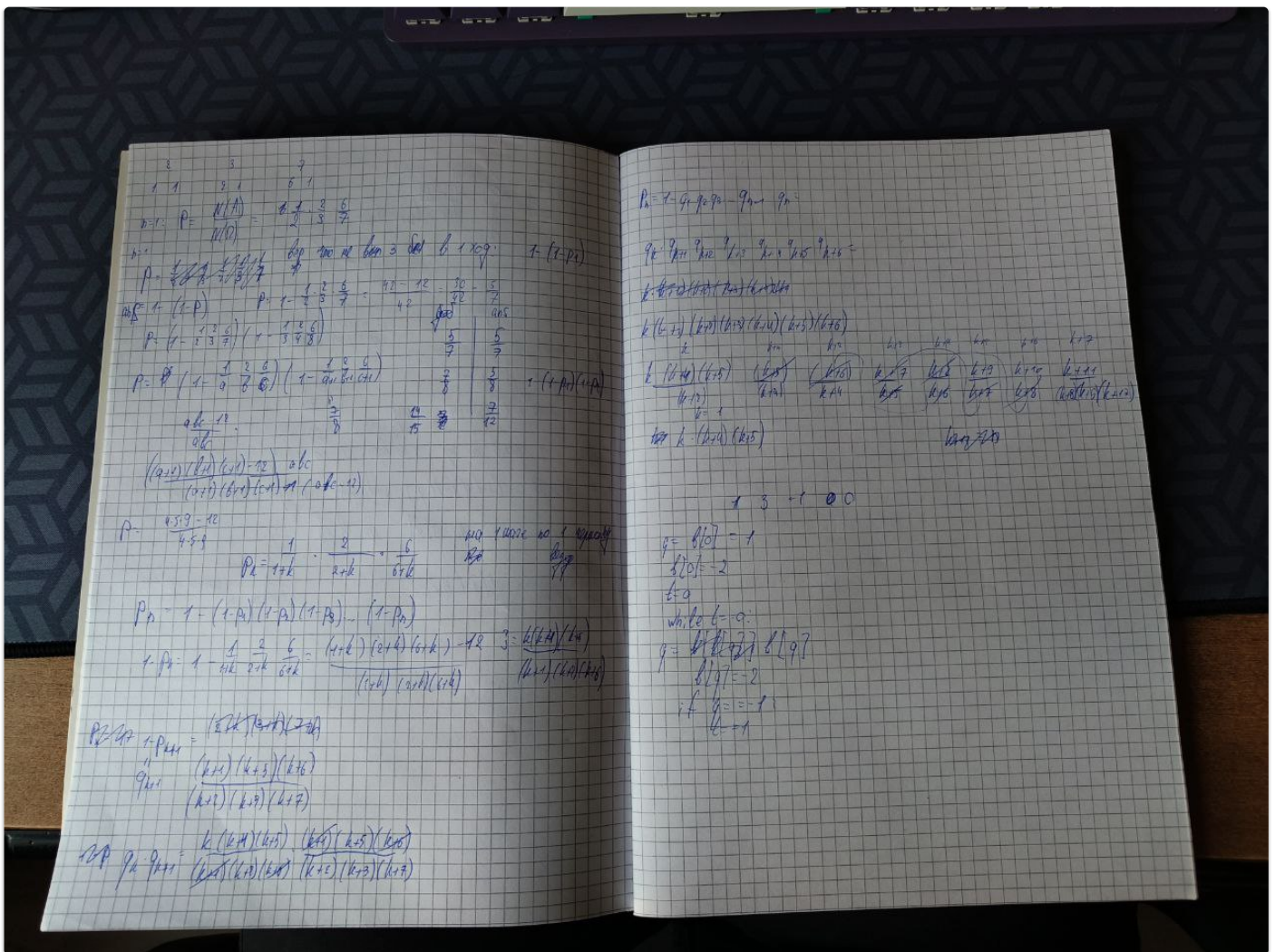
5

5 11

Пояснение

Моя самая любимая задача - задача на **Теорию Вероятностей!** Однако тут, в отличии от других задач, много кодить не получится, нужно считать...

Вот мои вычисления, был найден мною метод упрощения и сокращения **почти всех элементов произведения обратной вероятности**, что позволяет считать абсолютно для любого числа вероятность (проверял, 100000000 считается за 0.3 секунды, в ограничения заходим).



🕒 Что такое Fraction?

Это модуль для оптимального сокращения дробей до неделимых частей. И не надо говорить что нужно самому это делать, я кучу времени проёл на то чтобы придумать хоть что-то

```
from fractions import Fraction
n=int(input())
a=1
b=1
if n<=100:

    for i in range(1,n+1):
        if n==1:
            a*=i*(i+4)*(i+5)
            b*=(i+1)*(i+2)*(i+6)
        else:
            if (i==1):
                a*=i*(i+4)*(i+5)
                b*=(i+2)
            elif i==n:
                a*=(i+4)
                b*=(i+1)*(i+2)*(i+6)
            else:
                a*=(i+4)
                b*=(i+2)
    else:
        a=(n+4)*(n+3)*(n+2)
```

```
    b=2*(n+1)*(n+2)*(n+6)
a=b-a
# print(a,b)
q=Fraction(a,b)
print(q.numerator,q.denominator)

# a*=i*(i+4)*(i+5)
# b*=(i+1)*(i+2)*(i+6)
```

Задача 6

F. Дикие неончики

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Интернет-магазин "Дикие неончики" работает по следующей модели: в течение дня сайт интернет-магазина принимает заказы с доставкой товаров на завтра и на послезавтра. Вечером приём заказов на сайте завершается, и полученные заказы передаются на склад для сборки. Ночью склад собирает заказы, и к утру передаёт собранные заказы в доставку. Днём склад не работает.

Каждый день на сайт приходит ровно N заказов, причём для каждого поступающего на сайт заказа вероятность того, что этот заказ нужно будет доставить послезавтра, равна P . Склад каждый вечер берет в работу ровно N заказов, причём в первом приоритете берутся в работу уже просроченные заказы, во втором - заказы с доставкой на завтра, и в третьем приоритете - заказы с доставкой на послезавтра. Считаем, что все взятые в работу складом заказы успешно собираются к началу следующего дня. В начальный момент (утро первого дня) времени бэклог склада содержит K заказов, которые нужно собрать к завтра (они ещё не просрочены - их нужно собрать за ближайшую ночь). Определить мат.ожидание количества просроченных заказов склада за T суток работы. Под сутками работы понимается цикл, в котором сначала сайт интернет-магазина в течение дня принимает заказы, а затем склад ночью собирает заказы. Заказ считается просроченным, если он не собран складом к началу требуемого дня.

Пример 1

Ввод

2 0 2
0.5

Вывод

0

Пример 2

Ввод

2 1 2
0.5

Вывод

0.5

Пример 3

Ввод

5 1 1
0.125

Вывод

0.512908935546875

Формат ввода

В первой строке входных данных содержится три целых числа, разделенных пробелами: N , K , T таких что: $1 \leq N \leq 1000$, $0 \leq K \leq 1000$, $1 \leq T \leq 365$ Во второй строке входных данных содержится единственное вещественное число P , такое что $0 \leq P \leq 1$

Формат вывода

В качестве ответа вывести единственное вещественное число - математическое ожидание

- ☒ A. Лента событий SQL
- ☒ B. Код ревью
- C. Даркстор
- ☒ D. Случай в казино
- ☒ E. Игра с шариками
- F. Дикие неончики**

количества просроченных заказов за T дней работы, с точностью до 6 знаков после запятой.

Ну, эту задачу для грузчиков даже мне лень решать. Предлагаю кому-нибудь взять ее в качестве самостоятельного изучения, если кто-нибудь когда-нибудь отправит мне решение этой задачи, я лично вручу ему 100 рублей. В каком бы месте мира он не находился (ну, только договориться надо о встрече).

ИТОГО

- ☒ A. Лента событий SQL
- ☒ B. Код ревью
- ☐ C. Даркстор
- ☒ D. Случай в казино
- ☒ E. Игра с шариками
- ☐ F. Дикие неончики

Как видно, я решил всего лишь 4 из 6 задач, однако я считаю это очень и очень хорошим результатом. Ну и конечно же, SQLite3 сосёт бибу перед PostgreSQL))))))