

Требования к данному заданию:

- мы принимаем готовое тестовое задание только в формате ссылки на GitHub
- быть готовым на собеседовании ответить на любой вопрос по отправленному решению
- права на редактирование данного задания получить нельзя, запрос отправлять бессмысленно
- в конце заданию даны рекомендации для подготовки к собеседованию

Блок 1: Теория вероятности и логика

Задание 1: Фермер

На ферме содержатся шесть разных видов животных, и каждый раз, когда фермер заходит в сарай, он видит одно случайное животное. За день фермер заходит в сарай 6 раз.

Каково математическое ожидание количества разных видов животных, которые фермер увидит за день?

Ответ округлить до сотых, например: 4,12

3,99

Задание 2: Кулинарное соревнование

В конкурсе участвуют 80 шеф-поваров с уникальными уровнями мастерства. В первом этапе судьи случайным образом распределяют их по парам (в любом состязании двух шефов выигрывает тот, у кого выше уровень мастерства). На втором этапе шефы снова случайно образуют пары для финального раунда (пары могут повториться). Победную награду получают те, кто выиграл в обоих этапах.

Каково математическое ожидание числа победителей?

Ответ округлить до десятых, например: 33,5

28,6

Задание 3: Одинокая дорога

На пустынном шоссе вероятность появления автомобиля за 30-минутный период составляет 0.95.

Какова вероятность его появления за 10 минут? А за 27 минут?

Ответ дать в процентах, округлив до десятых через точку с запятой, например:

42,7; 95,0

63,1;93,3

Блок 2: Python

Задание 1: Изоморфизмы

Реализовать функцию (или тело функции), которая проверяет на изоморфность два слова. Пояснение: строки s и t называются изоморфными, если все вхождения каждого символа строки s можно последовательно заменить другим символом и получить строку t . Порядок символов при этом должен сохраняться, а замена — быть уникальной. Так, два разных символа строки s нельзя заменить одним и тем же символом из строки t , а вот одинаковые символы в строке s должны заменяться одним и тем же символом.

Пример:

```
s = 'paper'
```

```
t = 'title'
```

```
print(is_isomorphic(s, t))
```

Вывод:

```
True
```

Оценить оптимальность решения по времени и памяти и прикрепить текст кода.

Время $O(n)$

Память $O(n)$

```
def check_for_isomorphism(s: str, t: str) -> bool:
```

```
    if len(s) != len(t): return False
```

```
    index_of_chars_s = {}
```

```
    indexes_s = []
```

```
    index_of_chars_t = {}
```

```
    indexes_t = []
```

```
    for char in s:
```

```
        if char not in index_of_chars_s:
```

```
            index_of_chars_s[char] = len(index_of_chars_s)
```

```
            indexes_s.append(index_of_chars_s[char])
```

```
    for char in t:
```

```
        if char not in index_of_chars_t:
```

```
            index_of_chars_t[char] = len(index_of_chars_t)
```

```
            indexes_t.append(index_of_chars_t[char])
```

```
    return indexes_s == indexes_t
```

Задание 2: Натуральная последовательность

Реализовать функцию (или тело функции), которая находит единственное отсутствующее число из последовательности натуральных чисел $1, 2, \dots, n$.

Пример:

```
nums = [1, 2, 3, 4, 5, 6, 8, 9, 10, 11]
```

```
print(missing_number(nums))
```

Вывод:

```
7
```

Оценить оптимальность решения по времени и памяти и прикрепить текст кода.

Время $O(n)$

Память $O(1)$

```
def find_missing_number(nums):
```

```
    n = 1
```

```
while nums[n-1]==n:n+=1
return n
```

Задание 3: Факторизация

Реализовать функцию (или тело функции), которая при введении натурального числа n разбивает его на простые множители (представить его в виде простых чисел).

Пример:

$n = 56$

```
print(prime_factors(n))
```

Вывод:

[2, 2, 2, 7]

Оценить оптимальность решения по времени и памяти и прикрепить текст кода.

Время $O(n^{0.5})$

Память $O(\log n)$

```
def factorization(n: int) -> list[int]:
    if n<1:raise ValueError("Число должно быть натуральным!")
    multipliers=[]
    multiplier=2
    while multiplier*multiplier<=n:
        while n%multiplier==0:
            multipliers.append(multiplier)
            n//=multiplier
        multiplier+=1
    if n>1:
        multipliers.append(n)
    return multipliers
```

Блок 3: SQL

Задание 1: Абитуриенты

Есть таблица examination с двумя полями: id (id абитуриента), scores (кол-во набранных баллов дополнительного вступительного испытания от 0 до 100).

Требуется реализовать запрос, который создаёт колонку с позицией абитуриента в общем рейтинге.

SELECT

id,

scores,

RANK() OVER (ORDER BY scores DESC) AS position

FROM

examination;

Задание 2: FULL JOIN

Представьте две таблицы: первая содержит 30 строк, а вторая — 20 строк. Мы выполняем операцию FULL JOIN между ними.

Какой диапазон возможного количества строк может быть в результирующей таблице, если учесть, что ключи для соединения могут быть как полностью совпадающими, так и абсолютно уникальными?

[20,50]

Задание 3: Покупки

```
create table account
(
  id integer, -- ID счета
  client_id integer, -- ID клиента
  open_dt date, -- дата открытия счета
  close_dt date -- дата закрытия счета
)

create table transaction
(
  id integer, -- ID транзакции
  account_id integer, -- ID счета
  transaction_date date, -- дата транзакции
  amount numeric(10,2), -- сумма транзакции
  type varchar(3) -- тип транзакции
)
```

Вывести ID клиентов, которые за последний месяц по всем своим счетам совершили покупок меньше, чем на 5000 рублей.
Без использования подзапросов и оконных функций.

```
SELECT a.client_id
FROM account a
LEFT JOIN transaction t
  ON t.account_id = a.id
  AND t.transaction_date >= CURRENT_DATE - INTERVAL '1
month'
  AND t.type = 'buy'
GROUP BY a.client_id
HAVING COALESCE(SUM(t.amount), 0) < 5000;
```

Блок 4: Статистика и АБ-тесты

Задание 1: Воодушевленное руководство

Вы – аналитик компании Самокат (сервис по доставке продуктов на дом).
Команда решила протестировать гениальную идею замены транспортного средства доставщиков и провели АБ эксперимент в небольшом городке РФ. Результаты превзошли все ожидания: время доставки значительно снизилось в несколько раз! Руководству не терпится применить изменения по всей стране. Делаем? Выберите все верные утверждения:

Делаем. Только применяем изменения в таком же масштабе (количество транспортных средств с изменением), как в городе, где проводили тест.
Тест нерепрезентативен, поэтому результаты применять по всей стране нельзя.
Тест репрезентативен относительно своей генеральной совокупности: таких же небольших городов, можем применять только в подобных городах.

Наличие эффекта подтвердило потенциал идеи, поэтому сразу применять по всем городам не будем, но можем провести эксперимент в других городах.

2,4

Задание 2: Основной показатель в статистике

Что такое p-value в статистическом тестировании? Выберите одно верное утверждение:

Вероятность, что нулевая гипотеза верна.

Вероятность наблюдения такого или более экстремального результата при условии, что нулевая гипотеза верна.

Значение уровня значимости, при котором отвергается нулевая гипотеза.

Среднее значение выборки.

2

Задание 3: Параметрический тест

Вам необходимо провести аб-тестирование, целевая метрика - среднее число уникальных покупок на пользователя. Размер выборки велик (более 5 млн наблюдений в группах теста и контроля), значительные выбросы в выборках отсутствуют.

Можем ли мы применить t-критерий Стьюдента для проверки гипотезы о неравенстве средних в тестовой и контрольной группах при условии, что распределение уникальных покупок является логнормальным?

Да

Блок 5: ML Base

Задание 1: Пони тоже кони

Вас просят разработать модель, классифицирующую лошадок и пони. Вместо разработки вы нашли на GitHub две интересные модели и после прогона на ваших данных одна из них показала ROC-AUC=0.7, а другая ROC-AUC=0.1. Какую модель вы возьмете для дальнейшей работы и что будете с ней делать?

Ответ написать развернутый, но кратко, например: "первую, отниму от метрики 0.1 для корректировки точности"

Первую, диагностировать и оптимизировать

Задание 2: Ручной счёт ROC_AUC

Классификатор выдал следующие прогнозируемые метки класса и вероятности принадлежности к классу "1". На основе полученных данных рассчитайте метрику ROC_AUC. Тезисно опишите ход решения.

Ответ округлить до сотых, например: 4,12

Истинная метка класса	Порог классификации (0.6)	Оценка вероятности
1	1	0.95
0	1	0.9
1	1	0.85
0	1	0.8
1	1	0.75
1	1	0.7

1	1	0.65
1	1	0.6
0	0	0.55
0	0	0.5
0	0	0.45
1	0	0.4
0	0	0.35
0	0	0.3
0	0	0.25

0,75

7 объектов класса «1» и 8 объектов класса «0». Число пар объектов $7 \cdot 8 = 56$.

Число пар объектов, где вероятность класса «1» больше вероятности класса «0» $8+7+6+6+6+6+3=42$. $AUC=42/56=0,75$

Задание 3: Ручной счёт корреляции

Рассчитайте линейную корреляцию Пирсона, на основе данных.

Какой вывод можно сделать на основе полученного результата? Можно ли утверждать, что существует причинно-следственная связь между количеством чашек кофе, выпитых студентами в течение экзаменационного дня, и их итоговым баллом за экзамен?

Ответ округлить до сотых, например: 4,12

Число выпитых чашек кофе	Балл за экзамен
1	85
1	88
2	79
2	81
2	84
2	65
3	67
3	58
3	76
4	49

-0,84

Есть сильная отрицательная линейная статистическая связь.

Нет, корреляция - это статистическая мера, не доказывающая причинно-следственную связь.

Рекомендации по подготовке к собеседованию в случае успешного решения тестового задания

- Знать основы теории вероятностей;
- Знать основы математической статистики;
- Иметь практический навык реализации базовых запросов на SQL;
- Знать основы реализации алгоритмических задач на Python и оценки сложности алгоритмов;
- Приветствуется коммерческий опыт и знания в области планирования и проведения АБ-тестов.