Домашнее задание

Основы синтакиса

Всего баллов: 10

Бонусные баллы: 1

Эта домашка на «общее владение синтаксисом». Будет хорошо, если вы освоетесь с языком с уже знакомыми задачами, попадаете на какие-то грабли с *BorrowChecker*-ом, набьете шишек. Дальше подобных заданий практически не будет.

1. Макс

Баллы: 1

> Зло – это зло, – серьезно сказал Ведь'*Макс* — меньшее, большее, среднее – все едино, пропорции условны, а границы размыты. Я не святой отшельник, не только одно добро творил в жизни. И если приходится выбирать между одним злом и другим, я предпочитаю *большее*.

Напишите функцию `max(a: u32, b: u32) -> u32`, возвращающую ~~максимум двух чисел~~ зло, которое выберет *Макс*.

2. Сортировка

Баллы: 5

> Макс повзрослел, жизнь его потрепала, и требования заказчика к вашей программе изменились:
> - теперь Максу приходится иметь дело со многими злами
> - Герой изменил своим принципам, понял, что мир многограннее и для принятия взвешенного решения ему нужно внимательно изучить весь спектр возможных вариантов.
> - Чтобы подчеркнуть *вещественность* зла, Макс перешёл на тип `f64`.
>
> Поэтому теперь ему нужен тулинг *иного уровня*.

Напишите функцию, принимающую на вход `Vec<f64>`, сортирующую его по возрастанию. Сортировка может быть любая, даже работающая за $O(n^2)$. Встроенными методами сортировки пользоваться нельзя — да и просто так не получится, **Rust** знает, что среди `float`-ов бывают `NaN`(Not a Number), а они очень плохо сортируются. Но это часть спецификации `float`-ов во всех языках, ничего не поделать.

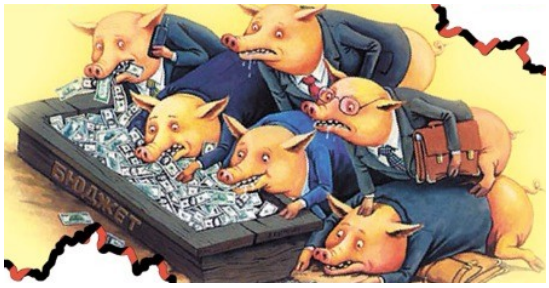
Ваша сортировка должна перемещать `NaN`-ы в конец массива.

P.S. Разумеется, встроенными (точнее, *обобщёнными* и *абстрактными*) методами воспользоваться можно, и мы это потом сделаем.

Мы дадим *1 бонусный балл* за $O(n \log n)$. Напишите это в комментарии, чтобы мы не пропустили.

3. Большой распил

Баллы: 4



Есть вероятность, что кто-то взял эту задачу для раунда CodeForces, ***слегка*** *адаптировав условия для конспирации*

> POV: вы с пацанами словили большой куш и уже готовы осваивать бюджет, но кое-что вас беспокоет: туманные обстоятельства и подгон данных в смете уже расследуют: Федеральная антимонопольная служба, Федеральная служба безопасноти, Генеральная прокуратура Российской федерации, а также СППК, FBI, KGB и прочие излишне любознательные огранизации, которые вечно суют нос не в своё дело.
>
> Ваша цель — прищемить их любопытные носы.
>
> К счастью, коллеги предлагают выход из этого затруднительного положения (они знают, что говорят — они так кучу раз делали!). Ищейки распределены равномерно по *«куску пирога»* (в коде — по одномерному массиву). Ключевая идея в том, что в силу дефицита кадров каждое ведомство выделило лишь по одному человеку, а люди из разных ведомств плохо уживаются друг с другом. Вы собираетесь *сыграть на противоречиях*: известно, что агенты с чётным ростом тепреть не могут «нечётников», и это отношение взаимно. Если чётные и нечётные агенты соберутся в закрытом помещении, они начинают обильно ругаться «стенка на стенку». Кроме того, если количество чётников и нечётников в помещении окажется одинаковым, они аннигируются. А противном случае — выживет хотя бы один и найдёт *изъян* в ваших схемах (вас бросит жена, вы сядете и... станете играть в теннис до скончания времён).
>
> Кроме того, чтобы разделить агентов на несколько секций, вы решили «распилить» пирог: построить перегородки между группами агентов на массиве. Перегородка не даёт агенту построить еуе-контакт с соседом. Очевидно, стоимость материала перегородки между агентами i и $i+1$ в этом случае $= |h_i - h_{i+1}|$, где h_i — высота глаз агента №i (на остальном пространстве можно сэкономить и не покрывать его перегородкой). Чтобы своре ищеек жилось несладко, вы решили построить как можно больше таких перегоордок (разумеется, необходимо уместиться в бюджет b).
>
> К сожалению, коллеги умеют программировать только на Python, а решение необходимо получить *blazingly fast*. Поэтому, вы приняли решение взять всё в свои ~~клешни~~ руки и написать программу для нахождения оптимального набора перегородок.
>
> Выделить, Освоить, Распилить

Вам дана последовательность целых чисел (высот глаз агентов), содержащая одинаковое количество чётных и нечётных чисел. Требуется в условиях ограниченного бюджета построить максимальное число перегородок, которые «распилят» последовательность (пирог) на непустые отрезки, на каждом из которых количество чётных чисел равно количеству нечётных чисел.

Перегородки разделяют последовательность на непрерывные подряд идущие отрезки, например:
`[4,1,2,3,4,5,4,4,5,5] → построили две перегородки → [4,1|2,3,4,5|4,4,5,5]`.

После разрезов на каждом отрезке количество четных элементов должно быть равно количеству нечётных.

Стоимость построения перегородки между агентами i и $i+1$ составляет $|h_i - h_{i+1}|$ у.е. Найдите максимальное количество перегородок, которые можно построить, потратив при этом не более b у.е.

Напишите функцию `saw(elements: Vec<u64>, b: u64) → usize`

Где целое число b ($1 \le b \le 100$) — объём бюджета, которым вы располагаете.

Вектор `elements` содержит n чисел $h_1, h_2, ..., h_n (1 \le h_i \le 100, 2 \le n \le 100)$ — уровни . Последовательность *содержит одинаковое количество чётных и нечётных элементов*.

Функция должна возвращать одно число — максимальное количество разрезов, которые можно провести, потратив не более b биткоинов.

Примеры:

```
b: 4
elements: 1 2 5 10 15 20
→ 1
```

```
b: 10
elements: 1 3 2 4
→ 0
```

```
b: 100
elements: 1 2 3 4 5 6
→ 2
```