

▼ Задание

Здесь вам нужно будет настроить работу дронов. Для этого нужно написать код в функциях ниже. Вы получаете баллы, если сможете пройти тесты и об этом будет явно написано.

Перед работой - создайте копию в гугл диске и работайте там. Для создания копии: Файл -> Сохранить копию на диске

Работа с окружением

Окружение называется `Google Collab` - это удобный способ быстро писать код на питоне и запускать его.

- Внутри одной ячейки вы можете писать любой код.
- Чтобы запустить его - наведите курсор на левый верхний угол ячейки и нажмите значок треугольник. Или просто внутри ячейки нажмите `Ctrl + Enter`

▼ Работа с заданиями и получение баллов

- Вам нужно будет реализовать 3 функции ниже, `count_energy`, `find_drone_with_most_energy` и `find_shortest_segment`.
- Что именно нужно написать и что вернуть - читайте в комментариях внутри функций и в тесте перед заданием.
- Параметры функции и их названия менять нельзя.
- При реализации функций убирайте строчку `raise NotImplementedError` - она не нужна.
- После реализации функции - запустите ячейку с тестами для функции (она идет сразу после ячейки с функцией), после вы получите набранное кол-во баллов.
- Вы можете свободно создавать новые ячейки (Вставка -> Кодовая ячейка), писать в них и запускать, не забудьте в конце удалить их (при нажатии на ячейку справа-сверху вы увидите значок корзины).

▼ Задание 1.

Чтобы запустить дроны - необходимо зарядить их. У вас есть текущий заряд дронов - он хранится в списке `energy_in_drone`. Также есть величина `energy_need` - необходимый заряд для дронов, чтобы они работали. Вам надо посчитать какое минимальное суммарное количество энергии надо потратить, чтобы довести каждого дрона до нужного уровня заряда.

Релизуйте для этого функцию `count_energy`. За это задание вы получите 1 балл.

Замечание: У некоторых дронов изначальный уровень заряда может быть выше, чем необходимый - в этом случае их заряжать не надо, они просто полетят с запасным зарядом.

```
def count_energy(energy_in_drone, energy_need):
    energy_extra = 0
    for i in energy_in_drone:
        energy_extra += max(0, energy_need - i)
    return energy_extra
```

```
energy_in_drone_test = [
    [1, 1, 1, 1],
    [1, 2, 3, 4],
    [10, 3, 54, 2],
    [0, 0, 0, 0, 0, 0, 0, 100],
    []
]
energy_need_test = [
    1,
    5,
    8,
    5,
    1000
]
energy_extra_test = [
    0,
    10,
    11,
    35,
    0
]
for energy_in_drone, energy_need, energy_extra in zip(energy_in_drone_test, energy_need_test, energy_extra_test):
    assert energy_extra == count_energy(energy_in_drone, energy_need), \
        "Ваша программа работает неправильно, попробуйте исправить ошибку и запустить программу еще раз"
print("Молодцы! За это задание вы получаете 1 балл")
```

Молодцы! За это задание вы получаете 1 балл

▼ Задание 2

В сети дронов необходимо выделить *центрального* - он будет передавать собранную информацию на сервер. Из-за лишней работы он будет тратить энергии больше, чем остальные. Поэтому было решено выбрать дрона с максимальным зарядом. Ваша задача среди всех дронов найти номер дрона с максимальным зарядом. Если таких несколько - выберете из дронов с максимальным зарядом тот, который имеет минимальный номер (чтобы инженеру не пришлось долго идти до него).

Вам дан список `energy_in_drone` - заряды дронов. Вам необходимо вернуть номер будущего *центрального* дрона. Нумерация дронов идет с 0.

Реализуйте функцию `find_drone_with_most_energy`. За это задание вы получите 1 балл.

```
def find_drone_with_most_energy(energy_in_drone):
    maxn = 0
    for i in range(len(energy_in_drone)):
        if energy_in_drone[maxn] < energy_in_drone[i]:
```

```

    maxn = i
    return maxn
energy_in_drone_test = [
    [3, 1, 5, 2, 6, 2, 10, 8],
    [4, 2, 1, 6, 3, 4, 6, 1],
    [0],
    list(range(10**5)) + list(range(10**5)),
    [-10**100, 10**100]
]
best_drone_test = [
    6,
    3,
    0,
    10**5 - 1,
    1
]
for energy_in_drone, best_drone in zip(energy_in_drone_test, best_drone_test):
    assert find_drone_with_most_energy(energy_in_drone) == best_drone, \
        "Ваша программа работает неправильно, попробуйте исправить ошибку и запустить программу еще раз"
print("Молодцы! За это задание вы получаете 1 балл")

```

Молодцы! За это задание вы получаете 1 балл

▼ Задание 3

Запускать всех дронов разом - идея не самая лучшая. Большое количество машин будет создавать слишком много помех. Хочется запустить лишь какую-то часть. Но если запустить дронов слишком мало - они не смогут выполнить свою работу. Также из-за технических характеристик вы можете запускать только группу подряд идущих дронов (дроны выстроены в ряд). Для каждого дрона вам известно его полезность (список `drone_utility`) - их посчитали инженеры за вас. Также вам известен необходимый уровень полезности (`min_utility`) - суммарная полезность запущенных дронов должна быть не меньше этого уровня. Найдите минимальное количество подряд идущих дронов, которых можно будет запустить, то есть суммарная полезность которых будет не меньше заданного уровня. Гарантируется, что суммарная полезность всех дронов не меньше необходимого уровня, а также, что все полезности неотрицательные.

Реализуйте функцию `find_shortest_segment`. Учтите, что количество дронов может быть до 200000. За это задание вы получите 1 балл. Большинство кода уже написано, вам лишь надо заполнить пропуски на месте `???`.

Замечание: для эффективного решения этой задачи мы воспользуемся методом 2-ух указателей. Его основная идея такова. Пусть у нас зафиксирована правая граница (r) группы дронов, которую мы хотим запустить. Как выбрать левую (l)? Если ее взять слишком далеко от r , то есть сделать группу очень большой, то суммарная полезность скорее всего будет достаточной, но количество дронов будет слишком большим. Если взять левую границу слишком близко к r , то количество дронов будет маленьким, что хорошо, но скорее всего нам не хватит суммарной полезности. Поэтому ответ находится где-то посередине, чем ближе мы двигаем l к r , тем лучше ситуация с количеством дронов, но тем хуже ситуация с суммарной

полезностью. Пусть мы нашли оптимальную l , то есть l - максимально возможная, чтобы полезности хватало. Что будет, если мы увеличим r на 1? Теперь суммарная полезность увеличилась и, возможно, мы сможем подвинуть l еще немного направо, потеряв какую-то лишнюю полезность. Поэтому идея алгоритма такая: перебираем r , при каждом увеличении r на 1 - двигаем l , пока позволяет суммарная полезность. Как только нашли оптимальную l для текущей r - учитываем эту группу в ответе. Из всех таких учтенных групп - выберем ту, в которой наименьшее количество дронов. Также для эффективности в переменной `current_sum` будем хранить суммарную полезность от l до r .

```
def find_shortest_segment(drone_utility, min_utility):
    l = 0
    current_sum = 0
    result = len(drone_utility)
    for r in range(len(drone_utility)):
        current_sum += drone_utility[r] # как измениться сумма полезностей, если добавить в группу
        if current_sum < min_utility:
            continue
        while l + 1 <= r and (current_sum - drone_utility[l]) >= min_utility: # мы хотим проверить,
            current_sum -= drone_utility[l] # как измениться сумма полезностей, если убрать из группы
            l += 1
        result = min(result, r - l + 1)
    return result
```

```
drone_utility_test = [
    [1, 1, 1, 1, 1],
    [1, 2, 3, 4, 5],
    list(reversed(range(100, 200))),
    [4, 1, 8, 10, 1, 12],
    list(range(10**5)) + list(reversed(range(10**5))),
    range(10**5),
    range(1, 10**5)
]
min_utility_test = [
    3,
    12,
    199,
    30,
    sum(range(10**5 - 5, 10**5)) * 2,
    sum(range(10**5)),
    sum(range(1, 10**5))
]
shortest_length_test = [
    3,
    3,
    1,
    4,
    10,
    10**5 - 1,
    10**5 - 1
]
for drone_utility, min_utility, shortest_length in zip(drone_utility_test, min_utility_test, shortest_length_test):
    assert find_shortest_segment(drone_utility, min_utility) == shortest_length, \
        "Ваша программа работает неправильно, попробуйте исправить ошибку и запустить программу еще"
```

```
print("Молодцы! За это задание вы получаете 1 балл")
```

Молодцы! За это задание вы получаете 1 балл

[Colab paid products](#) - [Cancel contracts here](#)

