Рекурсия

Рекурсия — это когда функция вызывает саму себя.

Пример:

Представь, ты стоишь перед лестницей с 10 ступенями.
Ты можешь сказать себе:

"Чтобы подняться на 10 ступеней, я сначала поднимусь на 1, а потом поднимусь на оставшиеся 9 тем же способом."

И вот ты говоришь то же самое на каждой ступеньке:

чтобы подняться на 9 — поднимусь на 1 и потом ещё на 8

потом на 8 — поднимусь на 1 и ещё на 7 и так далее...

Пока не дойдёшь до последней ступеньки, где скажешь:

"Если ступенек не осталось — я уже наверху, можно не звать себя больше." это условие остановки рекурсии aka базовый случай

Пример кода:

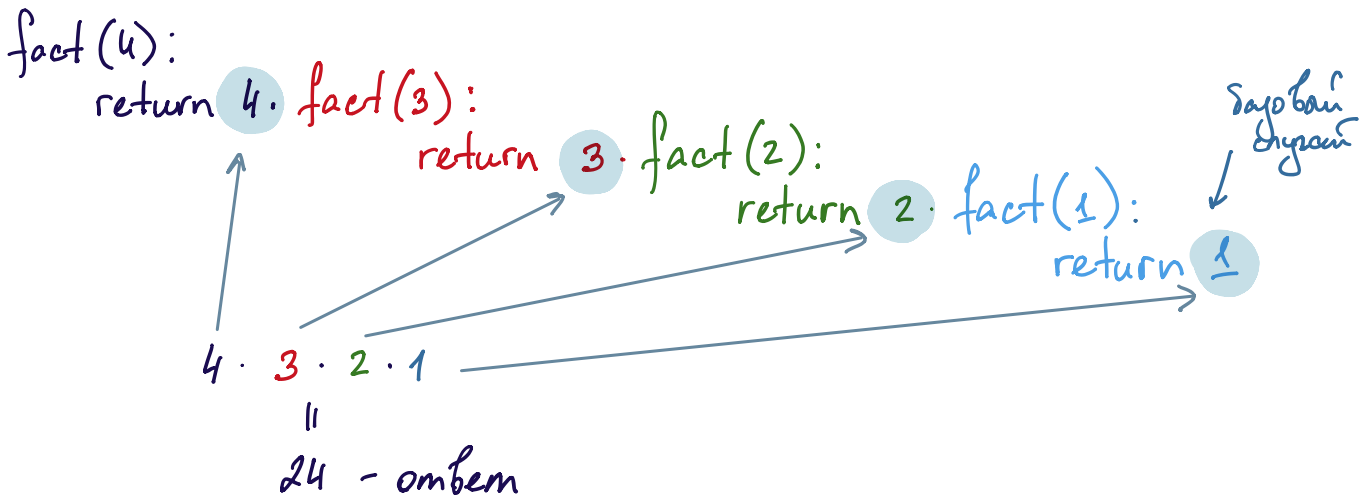
```python
def climb(n):
    if n == 0:
        print("Я уже наверху!")
        return
    print(f"Поднимаюсь на ступеньку {n}")
    climb(n - 1)
```

Вывод:

```
Поднимаюсь на ступеньку 3
Поднимаюсь на ступеньку 2
Поднимаюсь на ступеньку 1
Я уже наверху!
```

Разберем еще один пример: факториал

```python
def factorial(n):
    if n == 1:
        return 1
    return n * factorial(n - 1)
factorial(4)
```



Сортировка слиянием

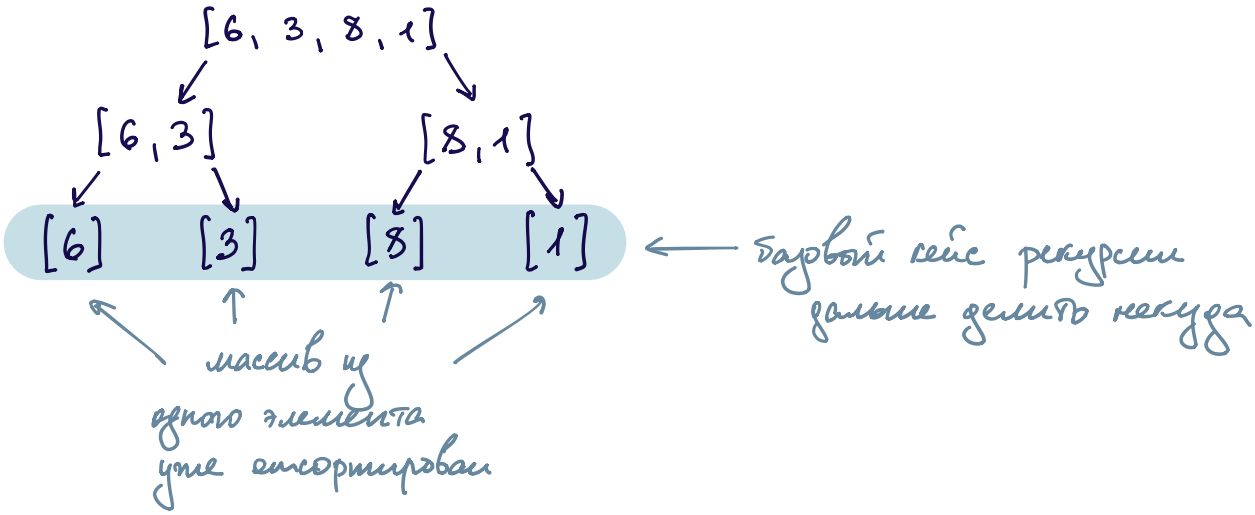
Сортировка слиянием работает по принципу разделяй и властвуй

Идея:

1. Раздели список на две половины
2. Отсортируй каждую половину (тем же способом — рекурсивно)
3. Объедини две отсортированные половины в одну

Пример: [6, 3, 8, 1]

1) Делим на 2 половины (до конца, рекурсивно)



2) Сливаем соседние массивы в один.

Будем по очереди сравнивать элементы массивов и каждый раз брать наименьший.

Пример: [1, 3, 5] [2, 4, 6] ! оба массива отсортированы

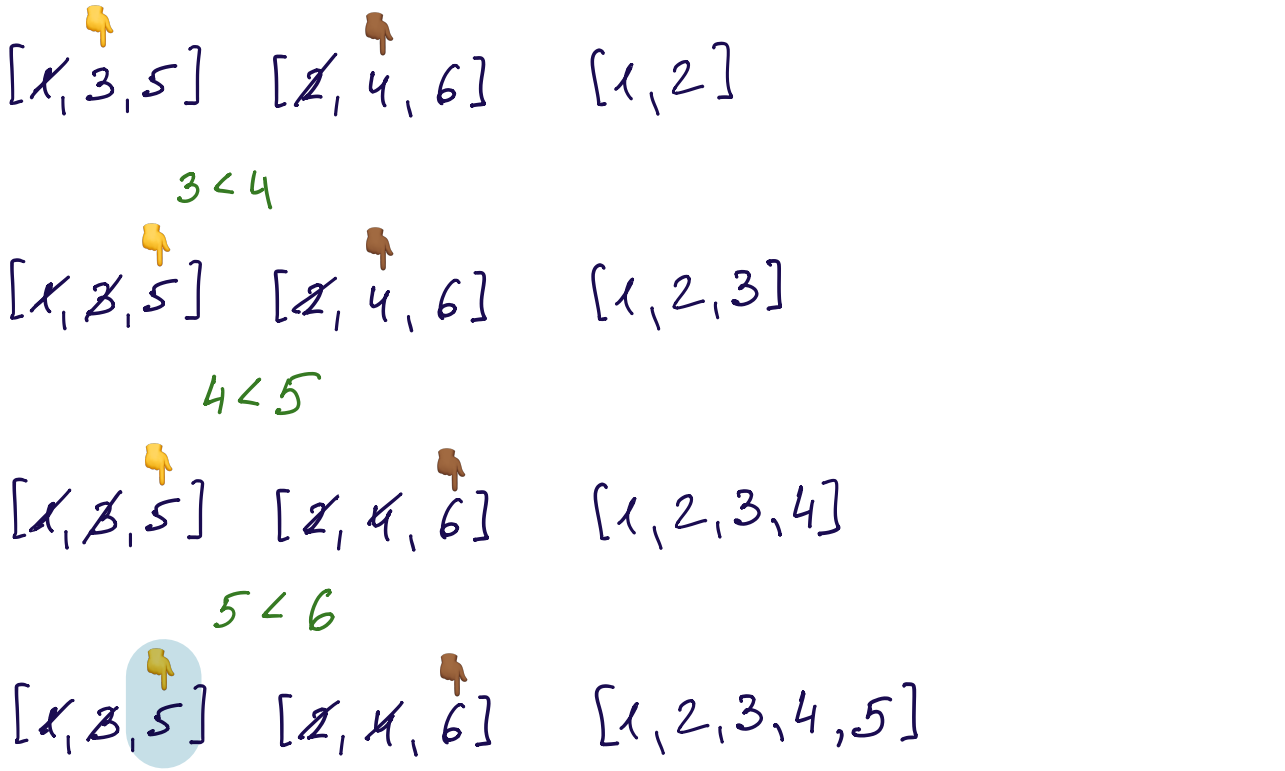
[] - результат слияния

[1, 3, 5] [2, 4, 6] [ ]

1 < 3 => добавляем 1, двигаем указатель первого массива

[x, 3, 5] [2, 4, 6] [1]

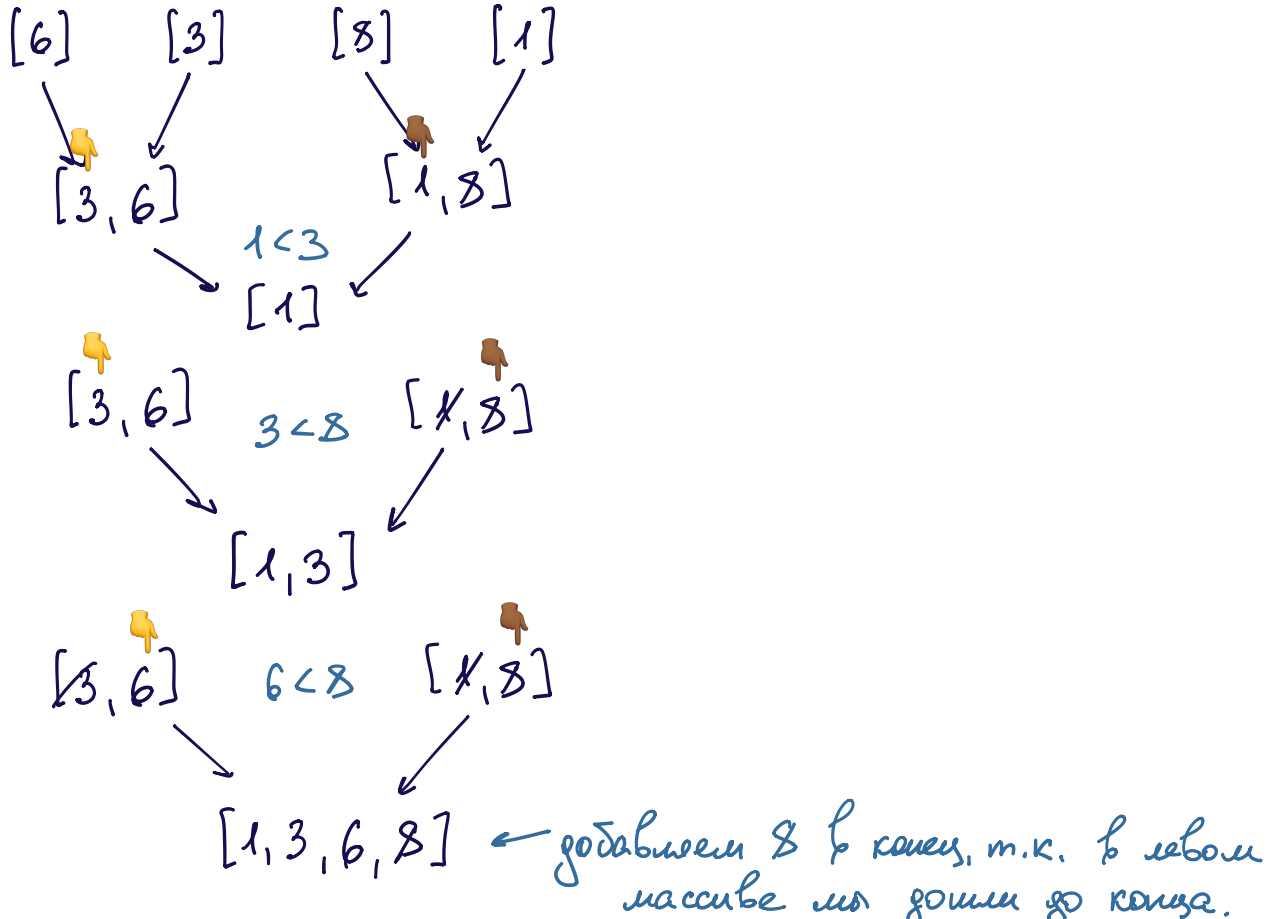
2 < 3 => добавляем 2, двигаем указатель второго массива



дошли до границы одного из массивов ⇒ добавляем оставшиеся элементы другого массива в конец.

[1, 3, 5] [2, 4, 6] [1, 2, 3, 4, 5, 6] итог.

Вернемся к нашему примеру.



← добавляем 8 в конец, т.к. в левом массиве мы дошли до конца.

```python
def merge_sort(arr):
    if len(arr) <= 1:
        return arr

    mid = len(arr) // 2
    left = merge_sort(arr[:mid])
    right = merge_sort(arr[mid:])

    return merge(left, right)


def merge(left, right):
    result = []
    i = j = 0

    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1

    result.extend(left[i:])
    result.extend(right[j:])
    return result

```

делим массив пополам рекурсивно

после разделения сливаем.

создаем отдельно функцию слияния

создаем указатели

если элемент левого массива < элемента правого добавляем левый и увел. указатель левого.

аналогично для правого.