

# Интенсив Python

## Лекция 8

### Ретроспектива по пройденным темам

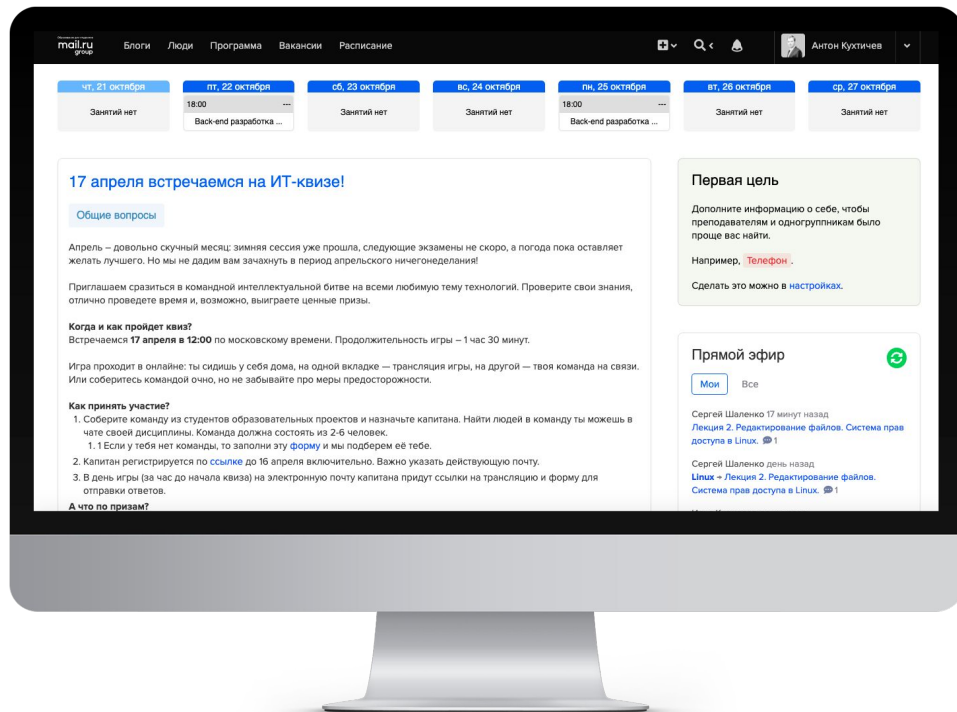
Кухтичев Антон



образование

# Напоминание отметиться на портале

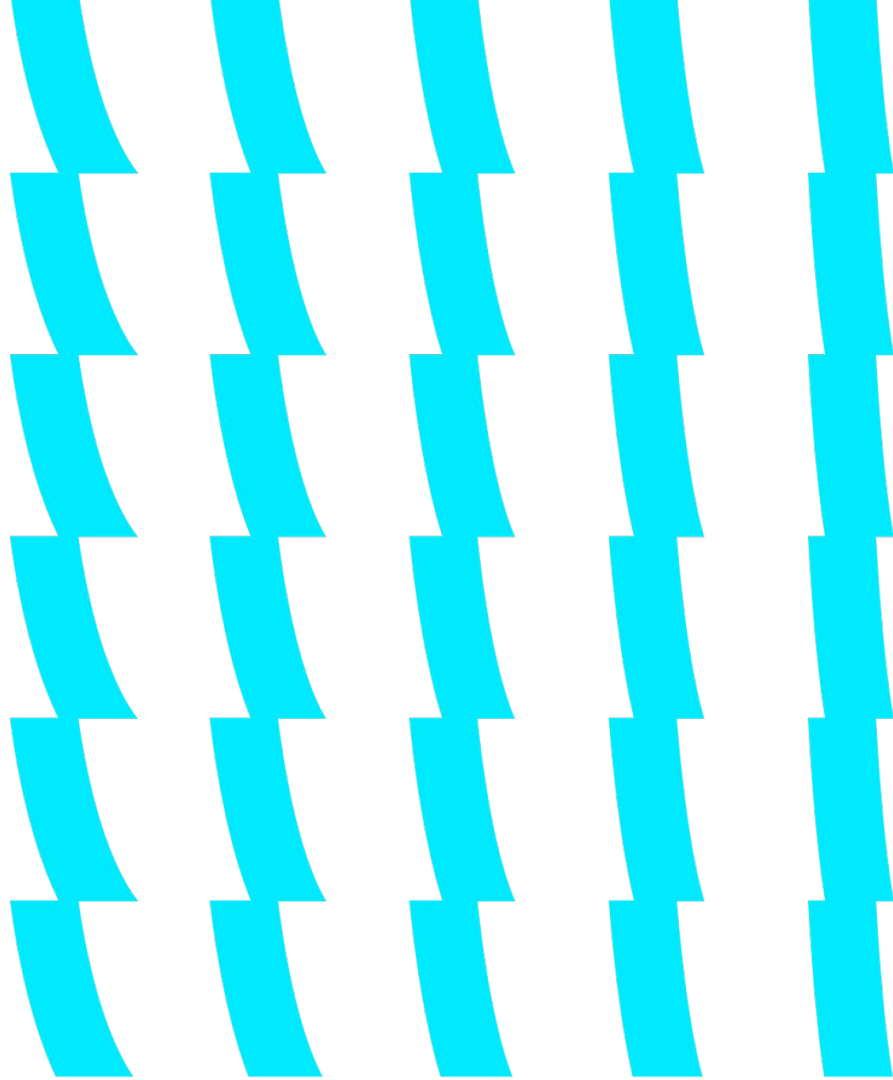
и оставить отзыв после лекции



## Содержание занятия

- Сложность алгоритмов
- Куча (heapq)
- Задача на heapq
- Задача на получение и парсинг xml и преобразование в json.

# Сложность алгоритмов



# O большое

«O» большое – математическое обозначение для сравнения асимптотического поведения алгоритма.

Фраза «сложность алгоритма есть  $O(f(n))$ » означает, что с ростом параметра  $n$  время работы алгоритма будет возрастать не быстрее, чем некоторая константа, умноженная на  $f(n)$ .

# О большое

Типичные значения:

1. Время выполнения константно:  $O(1)$
2. Линейное время:  $O(n)$
3. Логарифмическое время:  $O(\log n)$
4. Время выполнения « $n$  логарифмов  $n$ »:  $O(n \log n)$
5. Квадратичное время:  $O(n^2)$

# Примеры

1. `for i in range(n):`

`# do some work`

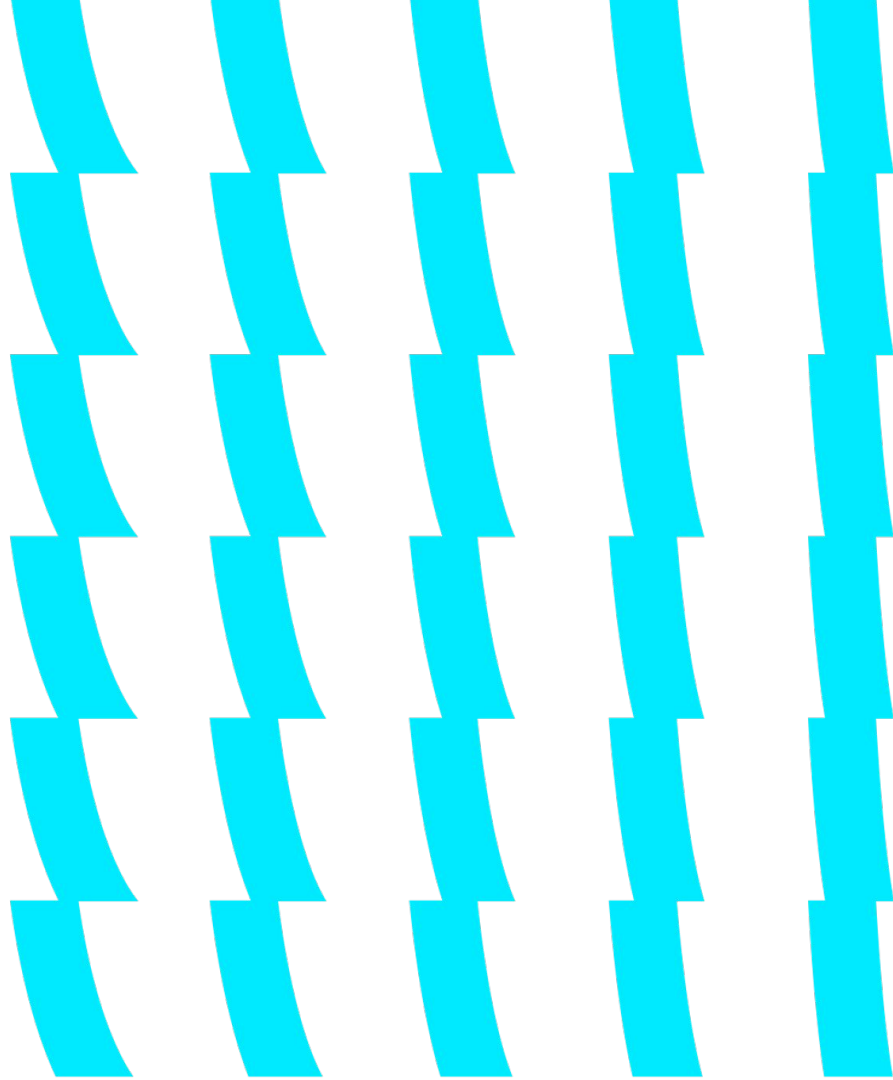
2. `for i in range(n):`

`for j in range(m):`

`# do some work`

3.

**heapq**





# heapq

`heappush(heap, item)`

`heappop(heap)`

`heappushpop(heap, item)`

`heapify(x)`

`heapreplace(heap, item)`

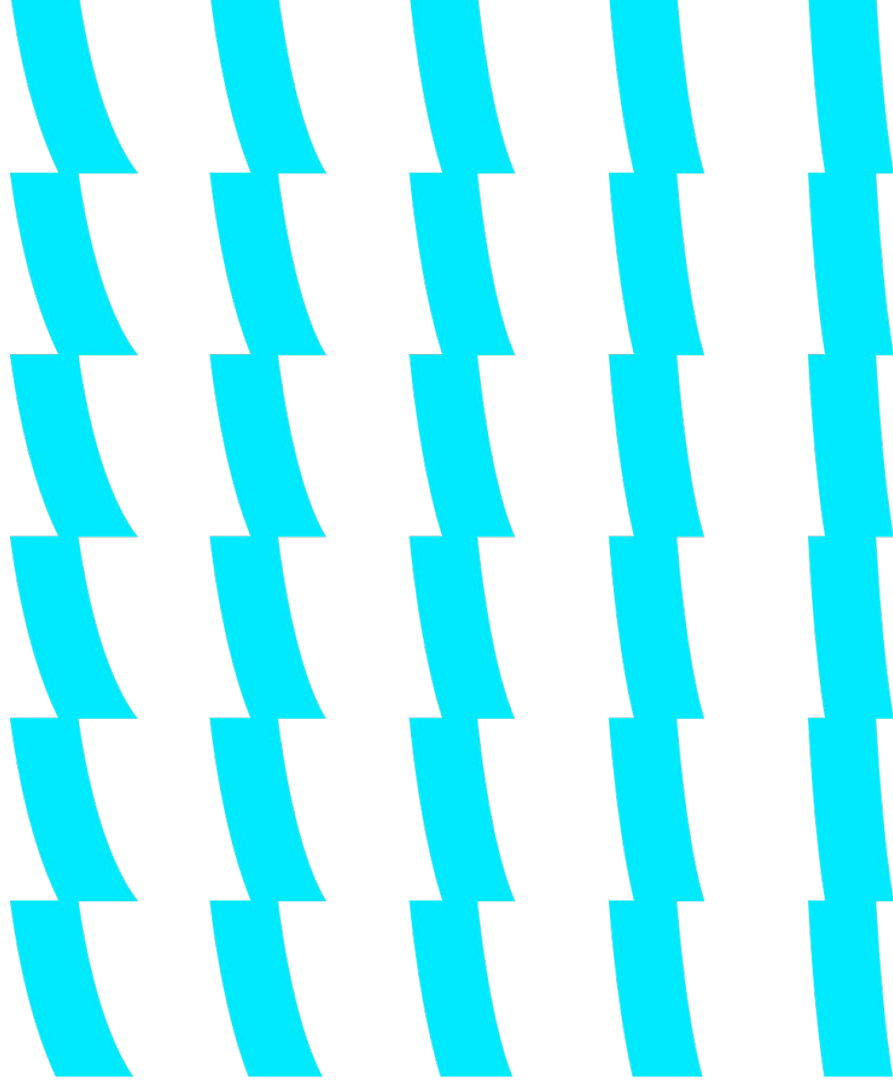
`merge(*iterables, key=None, reverse=False)`

`nlargest(n, iterable, key=None)`

`nsmallest(n, iterable, key=None)`

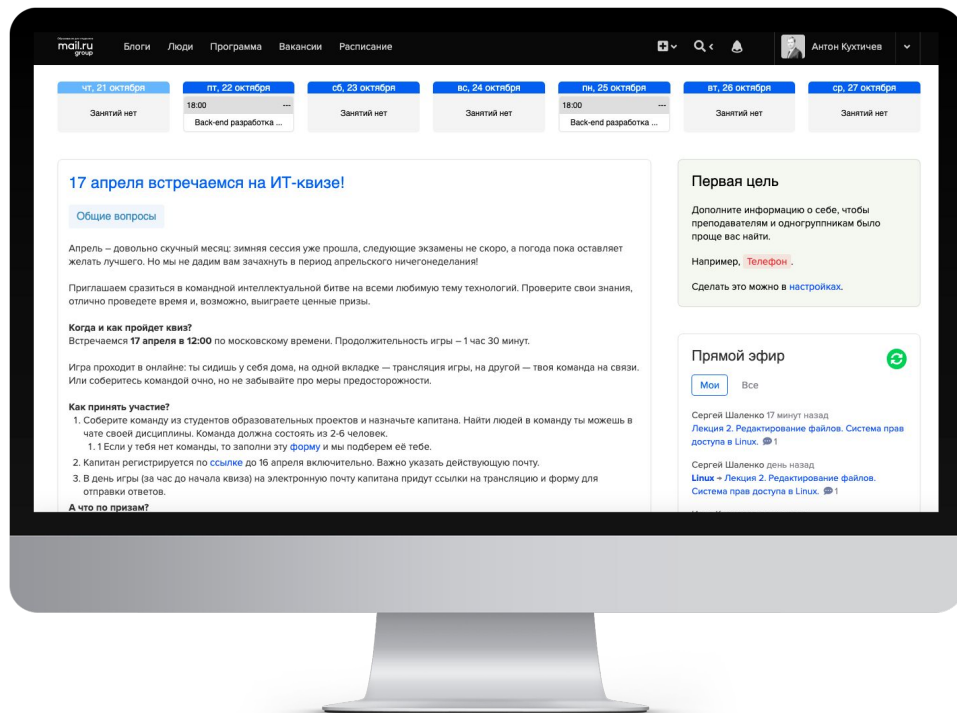
## Домашнее задание #8

- Реализовать функцию, принимающая на вход текст и образец. Нужно оптимально найти все позиции анаграмм образца в строке за оптимальное время.



# Напоминание оставить отзыв

Это правда важно



Спасибо за  
внимание

