

# Курс “Алгоритмы на python”

Занятие #7  
Куки

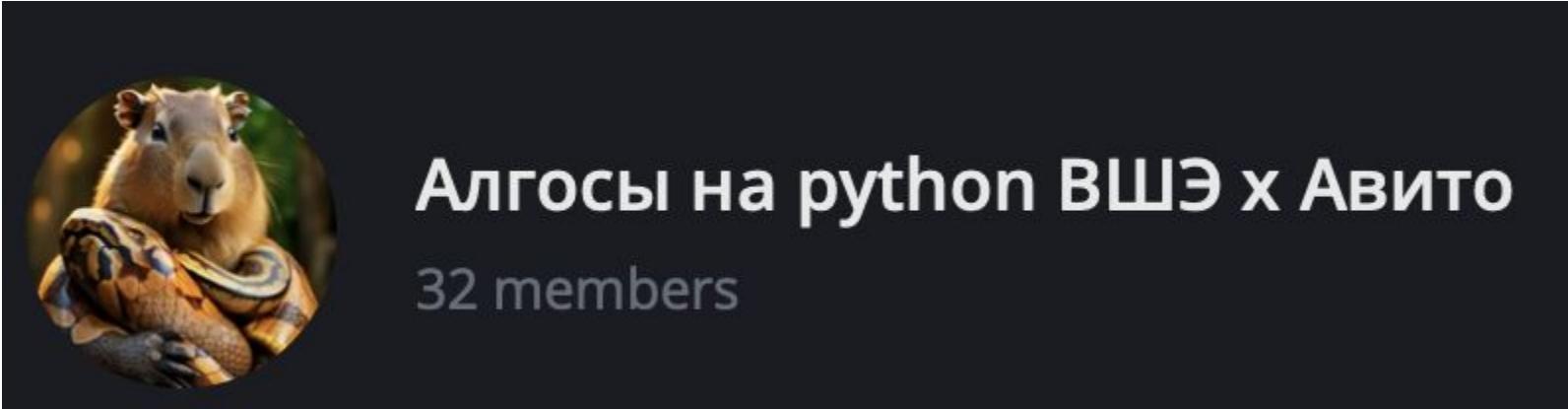
Сентябрь 2025



# **Единая точка входа/выхода – стек**

- <https://stepik.org/course/251189/>

# Вопросы и обсуждения – чат



# Посещаемость



# **Орг моменты**

# Дедлайны

3 дз:

- \* без штрафов 12 октября включительно
- \* минус балл – 26 октября включительно

4 дз

- \* без штрафов 20 октября включительно
- \* минус балл – 3 ноября включительно

5 дз

- \* без штрафов 29 октября включительно
- \* минус балл – 12 ноября включительно

6 дз

- \* без штрафов 5 ноября включительно
- \* минус балл – 19 ноября включительно

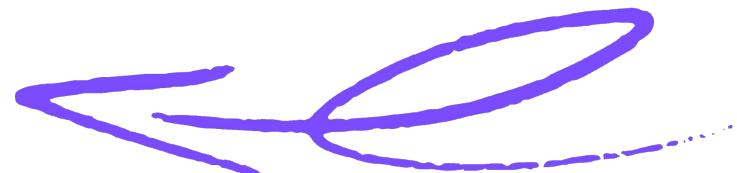
1 модуль

- Введение в алгоритмы
- Базовые структуры данных
- Хеш-таблицы
- Бинарные деревья поиска
- Рекурсия
- Сортировки

2 модуль

- Кучи
- Графы
- Динамическое программирование
- Алгоритмы в строках
- Алгоритмы в ML и LLM
- Итоговый контест

# Структура курса «Алгоритмы на питоне»



# План занятия



Часть I. Абстрактный тип данных: priority queue



Часть II. Куча (любая)



Часть III. Куча (бинарная)



Часть IV. Сортировка кучей



# Кучи



# Priority queue

`insert(key, value)` – добавить `value` с приоритетом `key`

`get_minimum()` – найти значение с **минимальным** приоритетом

`extract_minimum()` – найти и удалить

# Priority queue

Как реализовать такую структуру данных?

# Priority queue

Возьмем массив. Рисуем на доске.

# Priority queue

`insert(key, value)` –  $O(1)$

`get_minimum()` – найти значение с **минимальным** приоритетом –  $O(n)$

`extract_minimum()` – найти и удалить –  $O(n)$

# Priority queue

Возьмем массив + минимальное значение.  
Рисуем на доске.

# Priority queue

`insert(key, value)` –  $O(1)$

`get_minimum()` – найти значение с **минимальным** приоритетом –  $O(1)$

`extract_minimum()` – найти и удалить –  $O(n)$

# Priority queue

Сортированный массив. Рисуем на доске.

# Priority queue

`insert(key, value)` –  $O(n)$

`get_minimum()` – найти значение с **минимальным** приоритетом –  $O(1)$

`extract_minimum()` – найти и удалить –  $O(1)$

# Куча

Настало время кучи. Рисуем на доске.

# Куча

Свойство кучи:  
каждый элемент меньше своих детей.

# Куча

Операции:

get\_min

insert и sift\_up

extract\_min и sift\_down

# Куча

Как найти минимальный элемент?

# Куча

`get_min` – смотрим в корень –  $O(1)$

# Куча

Вставка

# Куча

insert и sift up –  $O(h)$

# Куча

extract\_min и sift down –  $O(h)$

# **Бинарная куча как массив**

Рисуем на доске

# Бинарная куча как массив

Храним как массив

`insert` нельзя куда угодно

`extract_min` нельзя через бесконечность  
(будут дырки)

# Heapsort

Как на основе того, что мы умеем делать, реализовать сортировку?

# Heapsort

**makeheap + N \* sift\_down**

**makeheap = N \* sift\_up**

**NlogN + NlogN = O(NlogN)**