

Углубленный Python

Лекция 14

Алгоритмы и структуры данных в Python

Кандауров Геннадий



education

Напоминание отметиться на портале

+ ОСТАВИТЬ ОТЗЫВ

vk образование

БлогиЛюдиПрограммаВакансииРасписание

+

Q

🔔

🔬

VK
Техно

▼

Открыт приём заявок!

чт, 8 сентября

Нет занятий

пт, 9 сентября

18:00 Углублённый Py...

СЗ

Введение в Python, основные понятия, тестирование

Г. Кандауров

сб, 10 сентября

Нет занятий

вс, 11 сентября

Нет занятий

пн, 12 сентября

Нет занятий

Углубленный Python

↓ 0 ↑

Блог для материалов по курсу "Углубленный Python"

57 читателей, 2 топика

Подписаться

Создать топик

Поиск по авторам, заголовку и тексту топика...

Найти

Добро пожаловать на курс!

Углубленный Python

Изменить

Удалить

Всем привет и добро пожаловать на курс по углубленному изучению Python!

Прямой эфир

МоиВсе

Геннадий Кандауров час назад

Углубленный Python → Добро пожаловать на курс! 0

Екатерина Черкасова 7 дней назад

Стажировка → Приглашаем мобильных, фронтенд- и бэкенд-разработчиков на Weekend Offer! 0

Дарья Вовченко 9 дней назад

Углубленный Python → Добро пожаловать в образовательные проекты VK Образование! 0

Дарья Вовченко 9 дней назад

Разработка веб-сервисов на

Содержание занятия

1. Основные понятия в алгоритмах
2. Бинарный поиск
3. Сортировки
4. Хеш-таблица
5. Куча

Основные понятия в алгоритмах



Алгоритмы и СД

Алгоритм представляет собой последовательность вычислительных шагов, преобразующих входные величины в выходные.

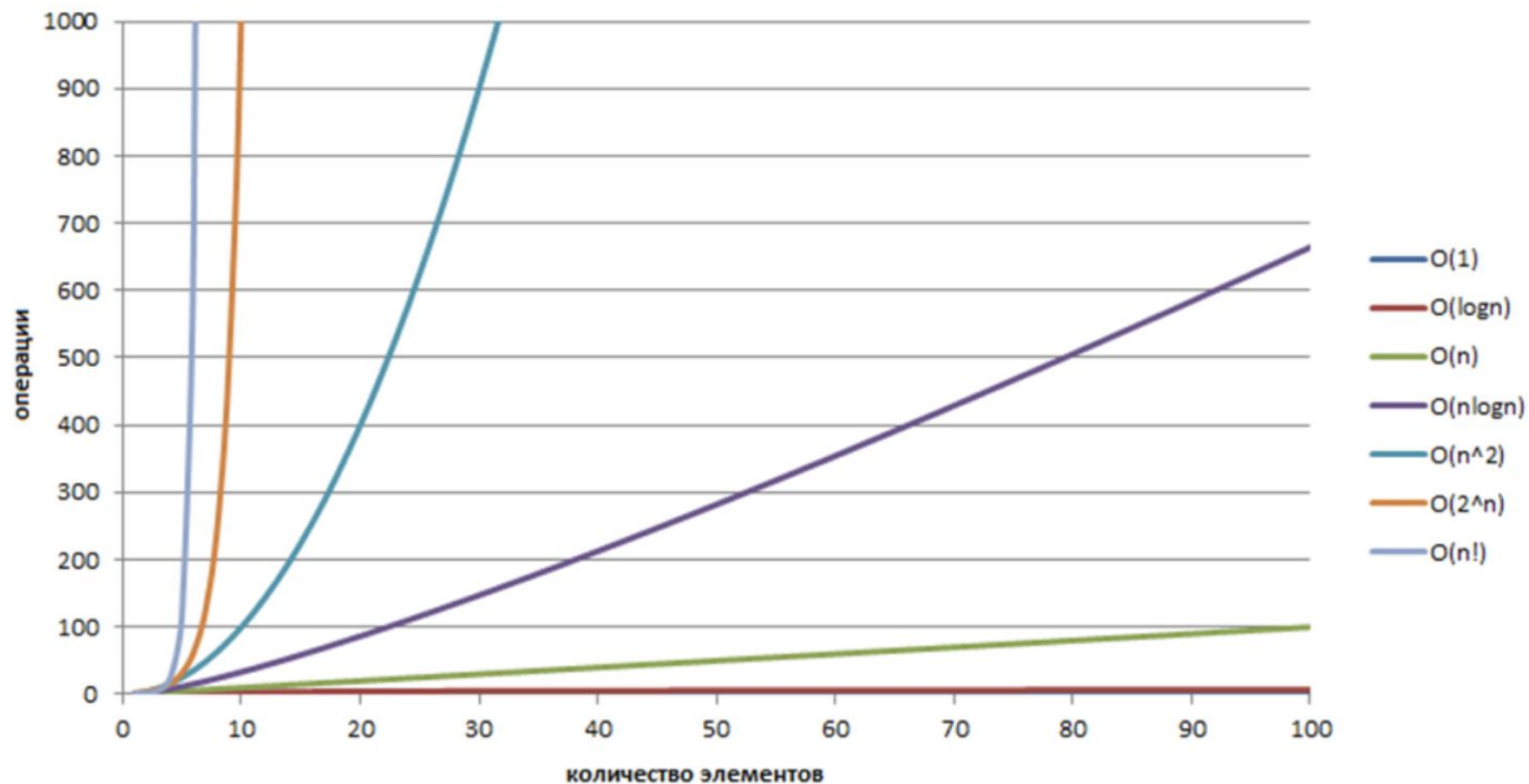
Структура данных - это способ хранения и организации данных, облегчающий доступ к этим данным и их модификацию.

O-большое

- Скорость алгоритмов измеряется не в секундах, а в темпе роста количества операции.
- По сути формула описывает, насколько быстро возрастает время выполнения алгоритма с увеличением размера входных данных.
- Время выполнения алгоритмов выражается как “O-большое”.
- Время выполнения $O(\log n)$ быстрее $O(n)$, а с увеличением размера списка, в котором ищется значение, оно становится намного быстрее.

Примеры

- $O(1)$
- $O(\log n)$
- $O(n)$
- $O(n \log n)$
- $O(n^2)$
- $O(n^3)$
- $O(e^n)$
- $O(n+k)$



Бинарный поиск



Бинарный поиск

- Алгоритм поиска элемента в отсортированном массиве, использующий деление массива на половины.
- Модуль bisect
 - bisect_left
 - bisect_right, bisect
 - insort_left
 - insort_right, insort

Сортировки



Сортировка вставкой

- Алгоритм:
 - Обработка элементов последовательности производится слева направо
 - Для очередного элемента выполняется поиск его места в отсортированной части массива.
 - Поиск осуществляется путем поэлементного сравнения справа налево
- Сложность:
 - $O(n)$ – в лучшем случае;
 - $O(n^2)$ – в среднем и в худшем

Сортировка вставкой

```
for j = 2 to A.length do
    key = A[j]
    i = j-1
    while (i >= 0 and A[i] > key) do
        A[i + 1] = A[i]
        i = i - 1
    end while
    A[i+1] = key
end
```

Сортировка слиянием

- Алгоритм:
 - Сортируемый массив разбивается на две части примерно одинакового размера.
 - Каждая из получившихся частей сортируется отдельно, например, тем же самым алгоритмом.
 - Два упорядоченных массива половинного размера соединяются в один.
- Сложность:
 - $O(n \log n)$

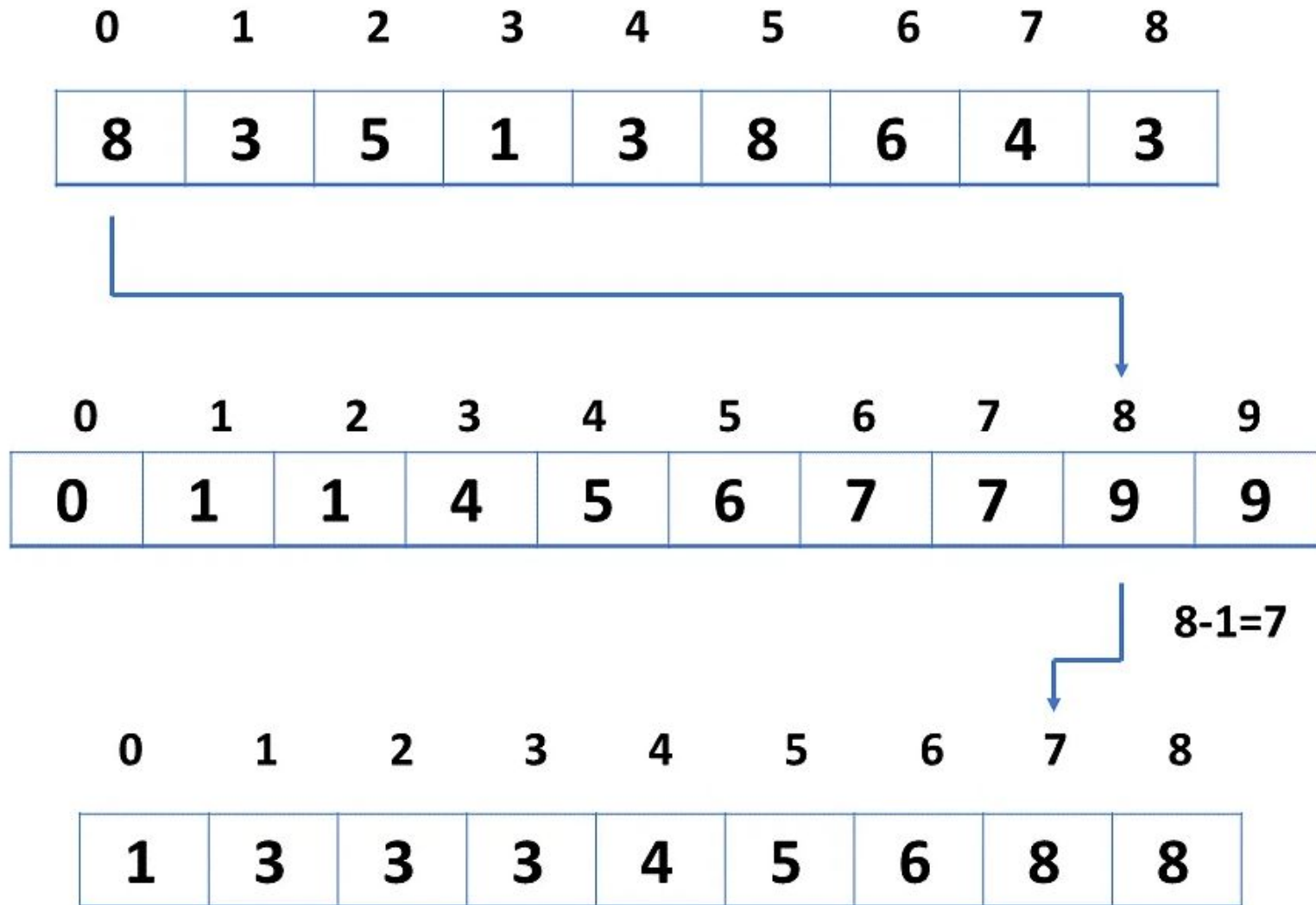
Сортировка слиянием

```
MergeSort(arr, left, right):  
    if left > right  
        return  
    mid = (left+right)/2  
    mergeSort(arr, left, mid)  
    mergeSort(arr, mid+1, right)  
    merge(arr, left, mid, right)  
end
```

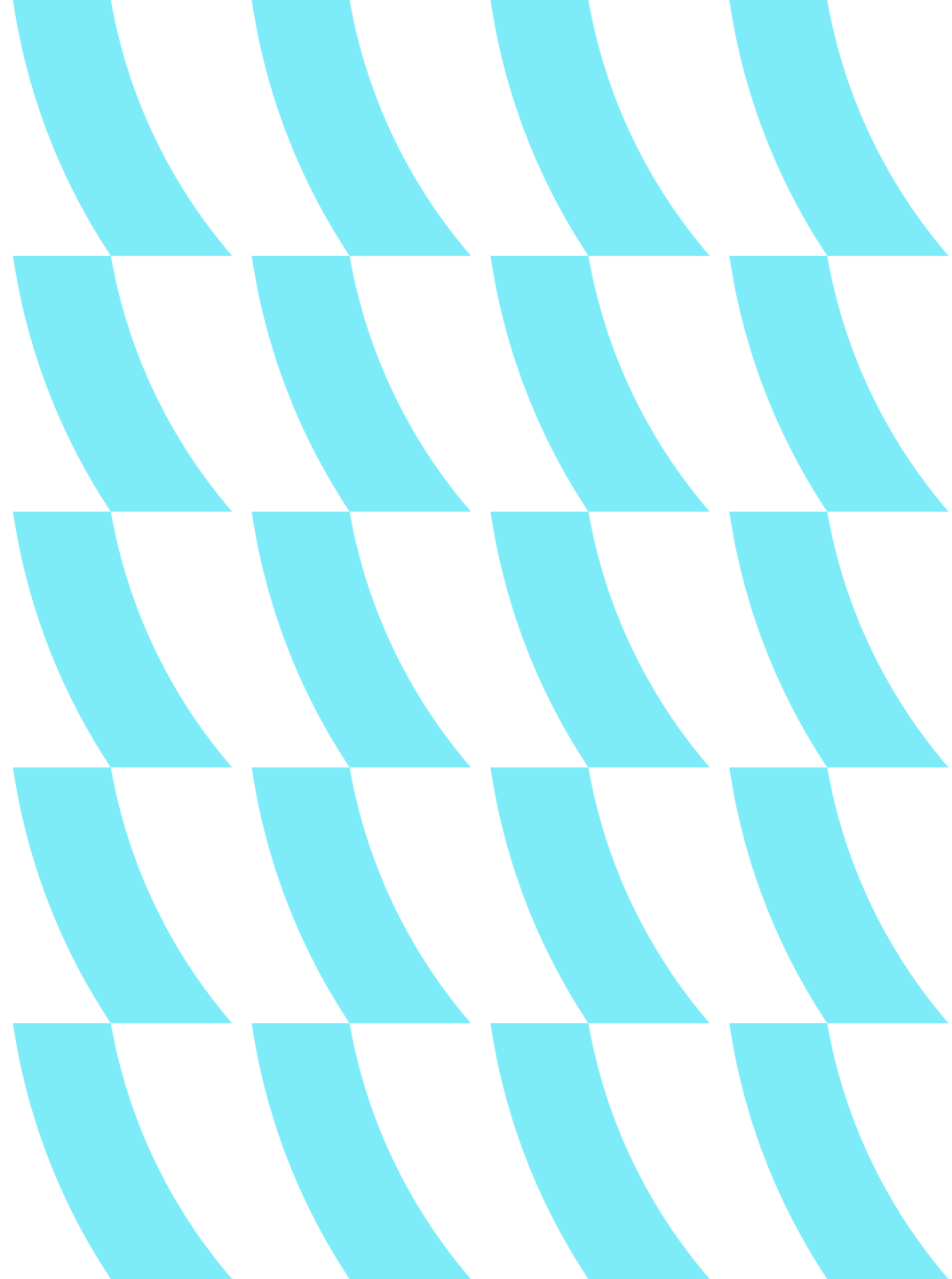
Timsort

- Эффективная комбинация нескольких других алгоритмов, приправленная собственными идеями;
- Алгоритм:
 - По специальному алгоритму разделяем входной массив на подмассивы.
 - Сортируем каждый подмассив обычной сортировкой вставками.
 - Собираем отсортированные подмассивы в единый массив с помощью модифицированной сортировки слиянием.
- Сложность: $O(n)$ в лучшем случае и $O(n \log n)$ в остальных

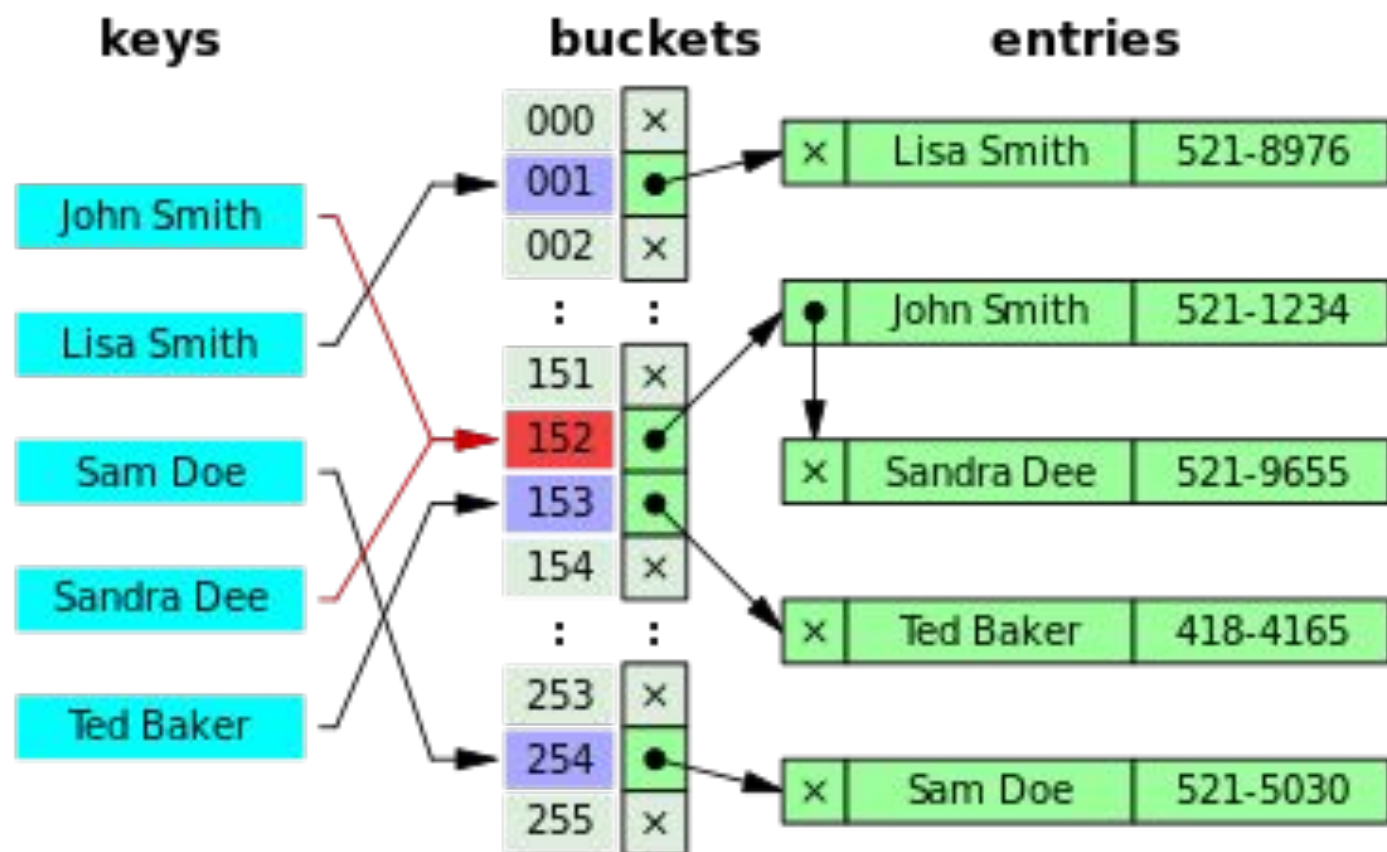
Сортировка подсчётом



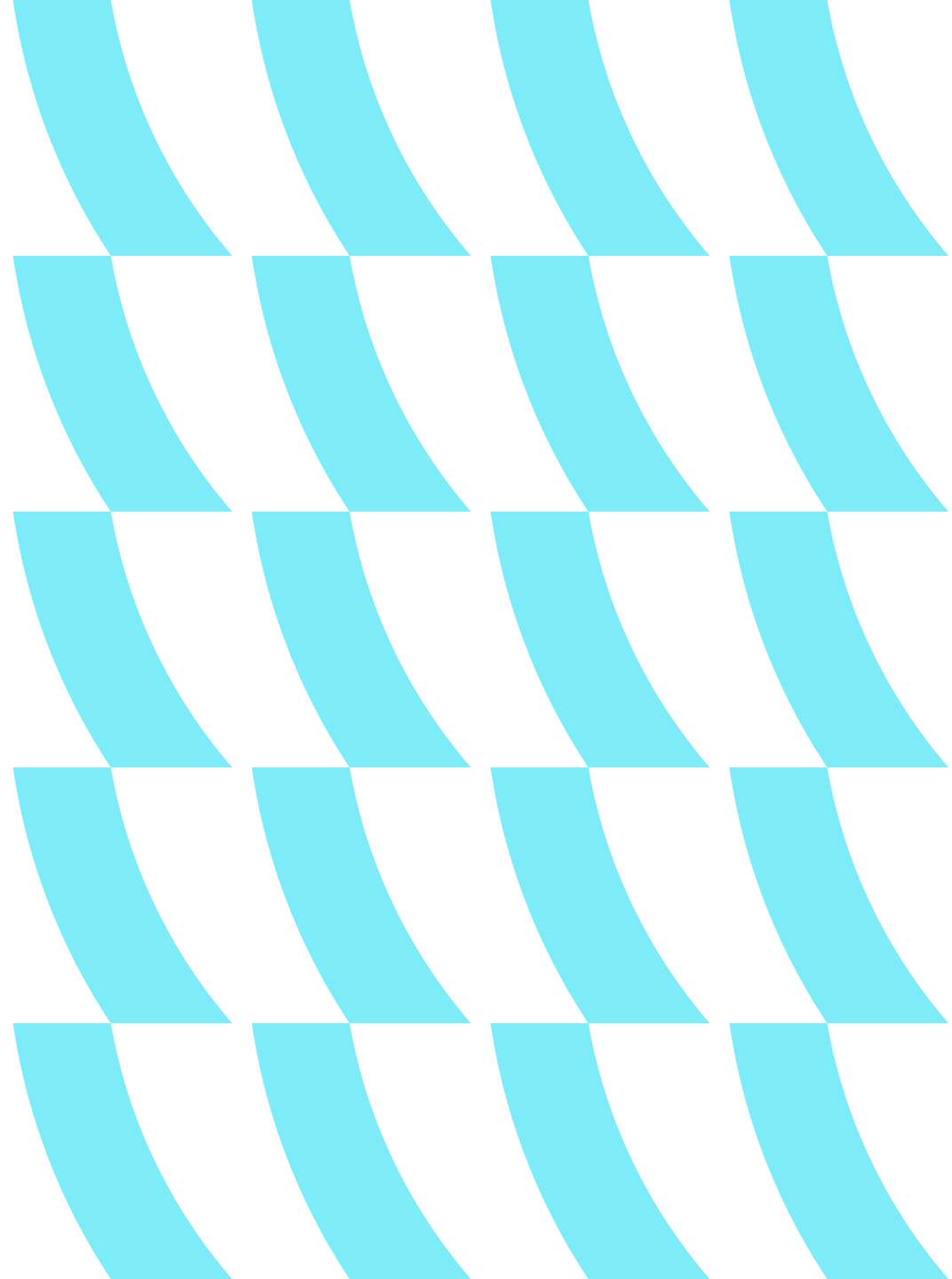
Хеш-таблицы



Хеш-таблицы



Куча



Куча

- **Пирамида** - это структура данных, представляющая собой объект-массив, который можно рассматривать как почти полное бинарное дерево;
- Каждый узел этого дерева соответствует определенному элементу массива;
- Пирамида характеризуется двумя атрибутами:
 - `length` - количество элементов массива
 - `heap_size` - количество элементов пирамиды
- В Python представлена в модуле `heapq`

Куча

Parent(i)

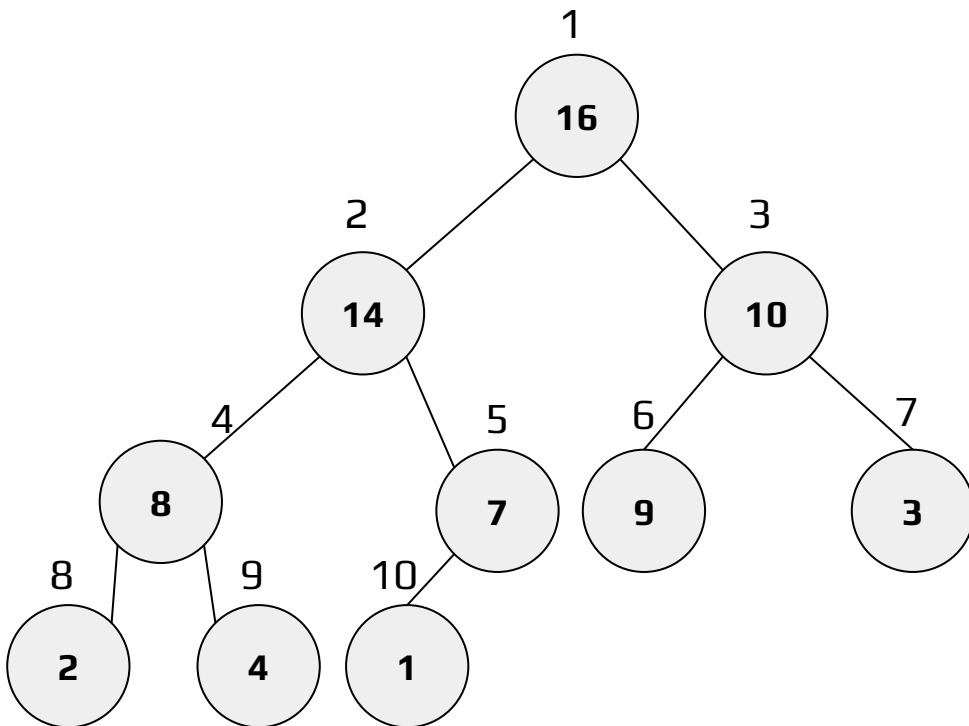
return $i/2$

Left(i)

return $2i$

Right(i)

return $2i + 1$



1	2	3	4	5	6	7	8	9	10
16	14	10	8	7	9	3	2	4	1

Куча

- `heapq.heappush(heap, item)`
- `heapq.heappop(heap)`
- `heapq.heappushpop(heap, item)`
- `heapq.heapify(x)`
- `heapq.heapreplace(heap, item)`

+ ОСТАВИТЬ ОТЗЫВ ПОСЛЕ ЛЕКЦИИ

Углубленный Python

↓ 0 ↑

Блог для материалов по курсу "Углубленный Python"

57 читателей, 2 топика

Подписаться

Создать топик

Поиск по авторам, заголовку и тексту топика...

Найти

Добро пожаловать на курс!

Углубленный Python

Изменить Удалить

Всем привет и добро пожаловать на курс по углубленному изучению Python!

Спасибо за
внимание

