

Python Введение в программирование

Gmail: vitalijfisenko5@gmail.com

Telegram: @vitaliifisenko

ВОПРОСЫ?)

ЗМЕЧАНИЯ

ВОПРОСЫ?)

Структуры данных

Что такое структура данных?

Структура данных — это контейнер, который хранит данные в определенном макете. Этот «макет» позволяет структуре данных быть эффективной в некоторых операциях и неэффективной в других.

Какие бывают?

Линейные - элементы образуют последовательность или линейный список, обход узлов линейен. Примеры: Массивы. Связанный список, стеки и очереди.

Нелинейные - обход узлов нелинейный, а данные не последовательны. Пример: граф и деревья.

Основные структуры данных

- Массивы
- Стеки
- Очереди
- Связанные списки
- Графы
- Деревья
- Хэш таблицы

Массивы

Самая простая и широко используемая структура данных. Другие структуры данных, такие как стеки и очереди, являются производными от массивов.

Основные операции:

- **Insert** вставляет элемент по заданному индексу
- **Get** возвращает элемент по заданному индексу
- **Delete** удаление элемента по заданному индексу
- **Size** получить общее количество элементов в массиве

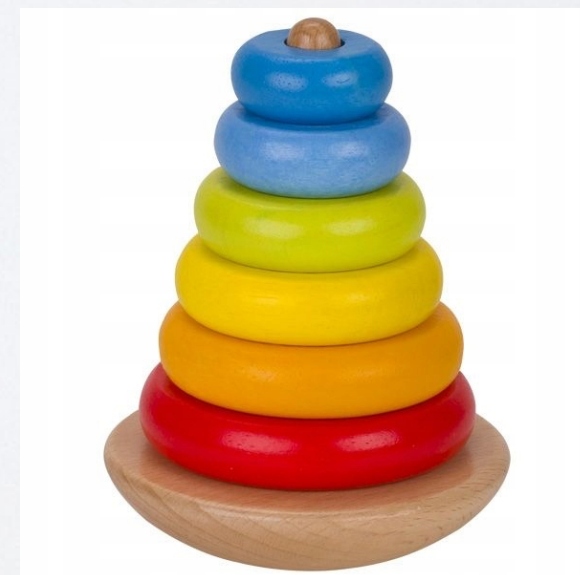
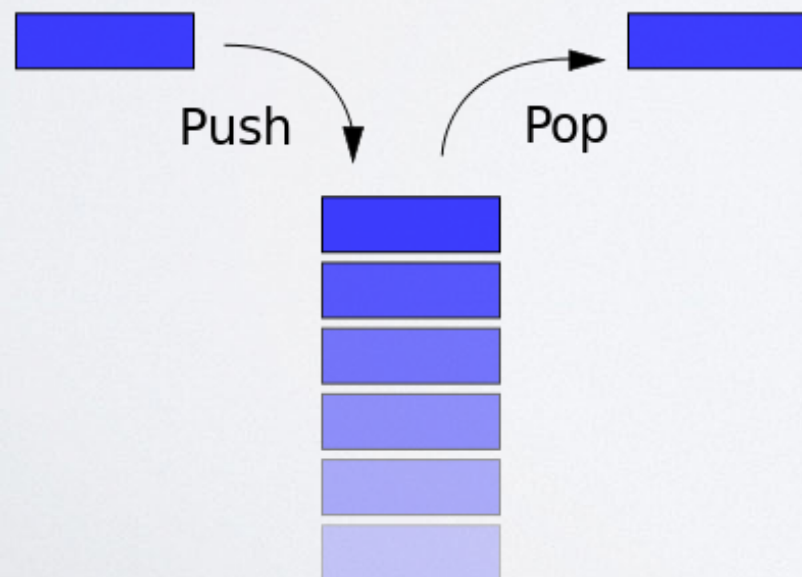


Стеки

Стек — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

Основные операции:

- **Push** вставляет элемент сверху
- **Pop** возвращает верхний элемент после удаления
- **isEmpty** возвращает true, если стек пуст
- **Top** возвращает верхний элемент без удаления из стека

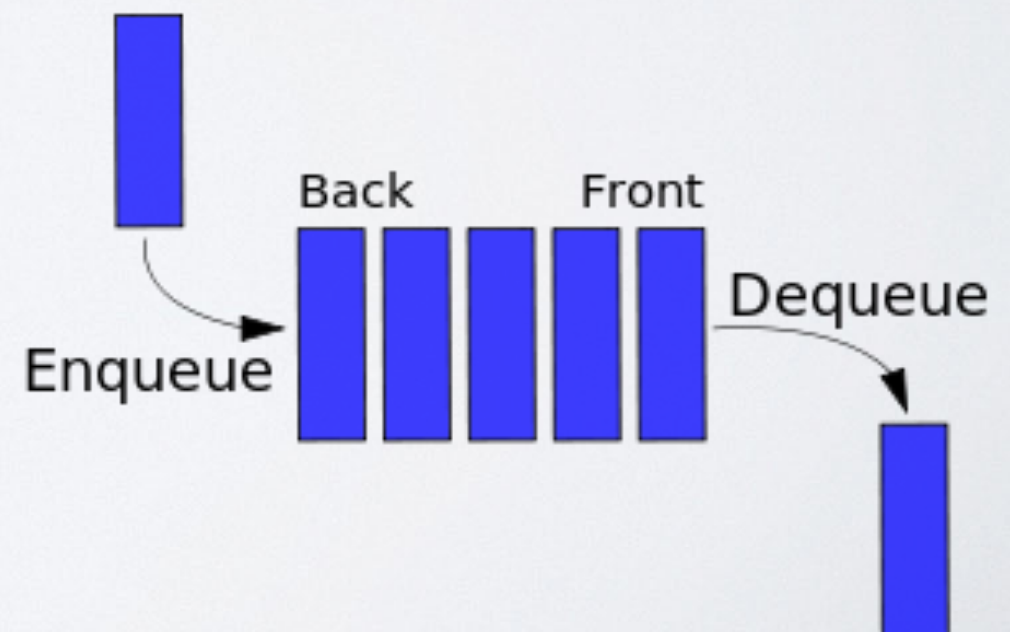


Очереди

Подобно стекам, очередь — хранит элемент последовательным образом. Существенное отличие от стека — использование FIFO (First in First Out) вместо LIFO.

Основные операции:

- **Enqueue** вставляет элемент в конец очереди
- **Dequeue** удаляет элемент из начала очереди
- **isEmpty** возвращает значение true, если очередь пуста
- **Top** возвращает первый элемент очереди



Связанный список

Связанный список – массив где каждый элемент является отдельным объектом и состоит из двух элементов – данных и ссылки на следующий узел.

Бывают

- **Однонаправленный** 1->2->3->4->NULL
- **Двунаправленный** NULL<-1<->2<->3->NULL
- **Круговой** 1->2->3->1

Основные операции:

- **InsertAtEnd** вставка заданного элемента в конец списка
- **InsertAtHead** вставка элемента в начало списка
- **Delete** удаляет заданный элемент из списка
- **DeleteAtHead** удаляет первый элемент списка
- **Search** возвращает заданный элемент из списка
- **isEmpty** возвращает True, если связанный список пуст

Хэш таблицы

Хэширование — это процесс, используемый для уникальной идентификации объектов и хранения каждого объекта в заранее рассчитанном уникальном индексе (ключе).

Объект хранится в виде пары «ключ-значение», а коллекция таких элементов называется «словарем». Каждый объект можно найти с помощью этого ключа.

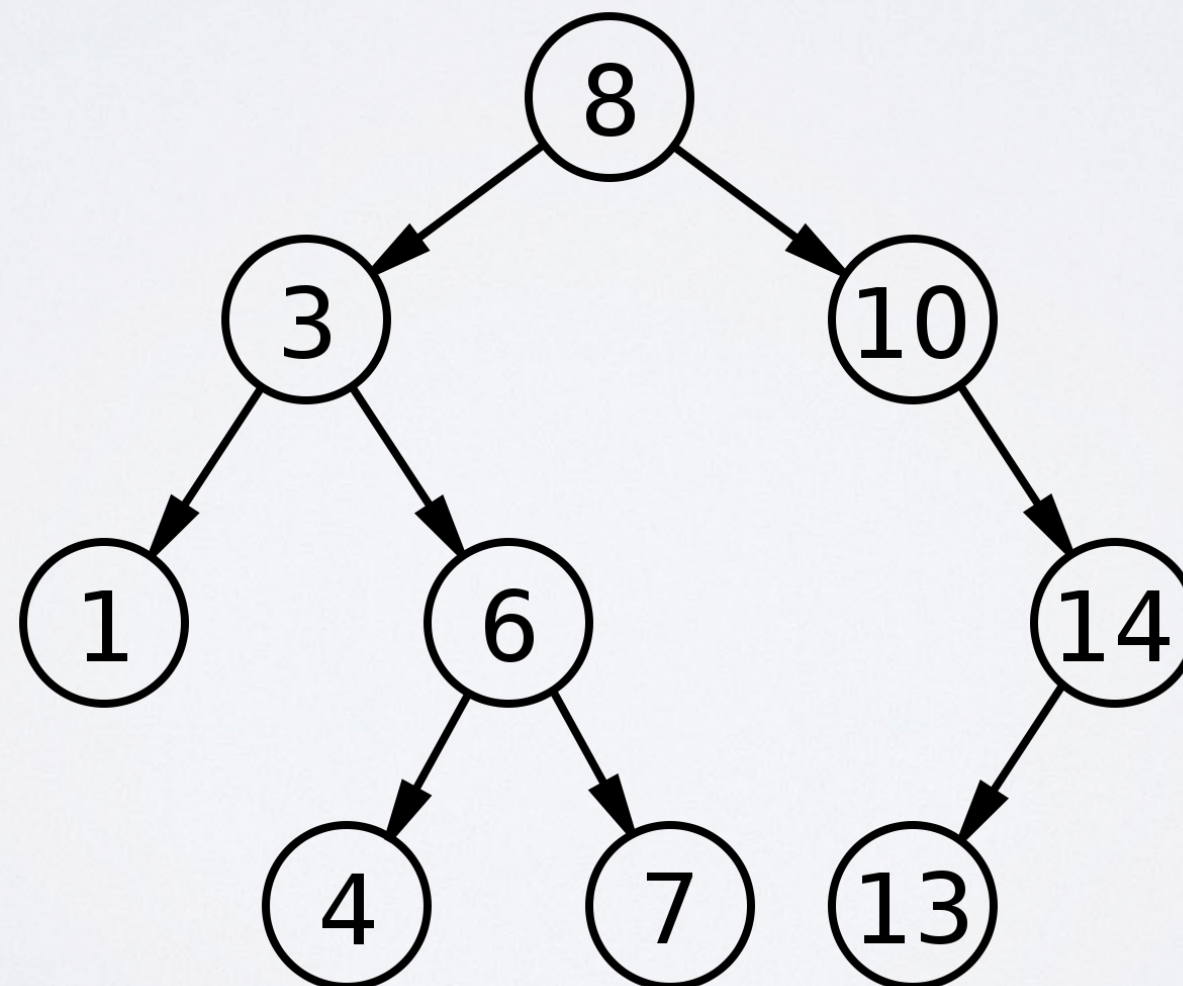
По сути это массив, в котором ключ представлен в виде хеш-функции.

| | | |
|----|-------|--------|
| | | |
| 3 | <key> | <data> |
| ⋮ | | |
| 16 | <key> | <data> |
| 17 | <key> | <data> |
| | | |

Деревья

Иерархическая структура данных, состоящая из узлов (вершин) и ребер (дуг). Деревья по сути связанные графы без циклов.

•



ПОЛЕЗНЫЕ ССЫЛКИ

<https://tproger.ru/translations/python-built-ins-worth-learning/>

<https://tproger.ru/translations/data-structure-time-complexity-in-python/>

<https://webdevblog.ru/bolshoe-o-cto-eto-takoe-pochemu-eto-vazhno-i-pochemu-eto-ne-vazhno/>

<https://habr.com/ru/company/ruvds/blog/515678/>

<https://habr.com/ru/post/188010/>

АЛГОРИТМЫ СОРТИРОВКИ