Основные структуры данных

1 Список

- Каждый элемент знает только свое значение.
- Каждый элемент знает где находиться следующий элемент списка.
- * В случае двусвязного списка каждый элемент знает где находится предыдущий элемент.

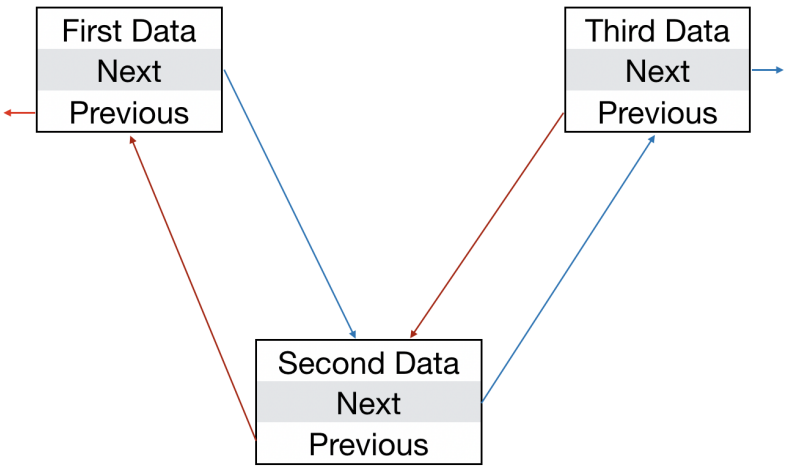

Рис. 1: Изображение списка

На рис. 1 изображена общая структура списка. Если на рис. 1 оставить только синие линии то получим простой список, если же оставить все линии, то получаем двусвязный список.

Основные операции:

- append() — добавить в конец списка,
- pop(index) — вытащить элемент со списка с индексом index,
- len() — возвращает длину списка.

2 Стэк

- Главное свойство, кто первый попадает в стэк, тот последним из него выходит.

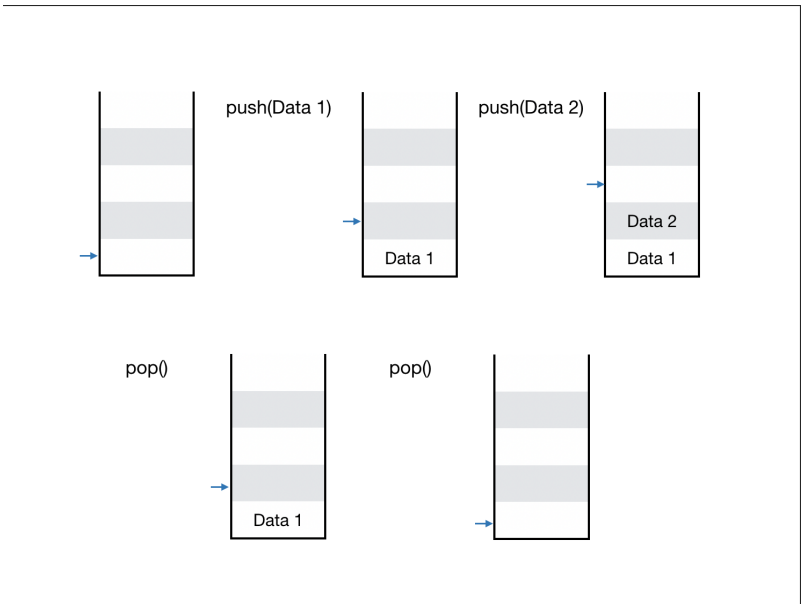

Рис. 2: Изображение стэка

На рис. 2 изображена общая структура стэка. Как видно сначала из стэка достается элемент который в него был поставлен в самом конце и последним достается элемент который был поставлен первым. На рис. 2 синяя стрелочка указывает, на место куда будет вставлен следующий элемент и какой элемент будет изъят следующим.

Основные операции:

- push() — добавить в вверх стэка,
- pop() — вытащить верхний элемент стэка,
- len() — возвращает глубину стэка.

3 Очередь

- Главное свойство, кто первый попадает в очередь, тот первым из нее выходит в отличии от стэка.

На рис. 3 изображена общая структура очереди. Синяя срелочка показывает в какое место будет вставлен следующий элемент, а красная стрелочка указывает откуда будет браться следующий элемент. Как видно, первый элемент который попал в очередь первый из нее выходит.

Основные операции:

- push() — добавить в конец очередь,
- pop() — вытащить первый элемент очереди,

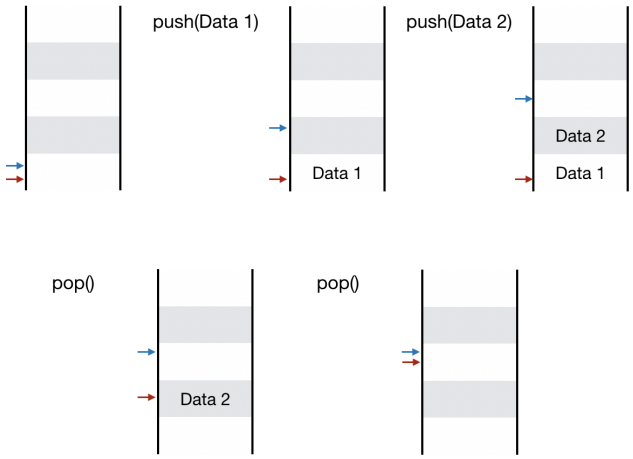

Рис. 3: Изображение стэка

- len() — возвращает длину очереди

4 Set

- Это структура данных, которая отвечает такой математической структуре как множество.
- Главная задача этой структуры, это ответить на вопрос есть ли элемент в множестве.
- В отличии от структур 1—3 эта структура не имеет порядка.

Основные операции:

- insert() — добавить элемент в множество,
- find() — найти элемент в множестве,

Операция find() вернет либо 0 либо 1 в зависимости от того, принадлежит элемент множеству или нет.

5 Map

- map и set очень похожие структуры данных.
- map отличается от set, только тем, что в нем храниться не только информация есть ли элемент а еще и каждому элементу ставиться в соответствии еще один элемент.
- Каждая пара состоит из key и data, где key должен быть уникальным.

Пример использования map очень просто, например мы хотим сделать перепись книг в библиотеке. В качестве уникальных key будет выступать название книги, а в качестве data будет выступать числа книг соответствующих этому названию.

6 Hash

- Хэш таблица — способ хранения данных
- Нужна hash функция для данных которые вы собираетесь хранить

Хэш таблица это просто массив фиксированного размера в котором хранятся некоторые данные, но доступ к этим данным происходит не по простой индексации, а при помощи hash функции.

Hash функция это некоторое отображения данных которые нужно сохранить в натуральное число — индекс массива.

Основной минус, что количество данных обычно много больше чем размер массива, поэтому на самом деле элемент массива это список тех элементов у которых Hash функция дает один и тот же результат. Список решает проблему так называемой коллизии.