53. Структуры данных и алгоритмы

Виды тестирования:

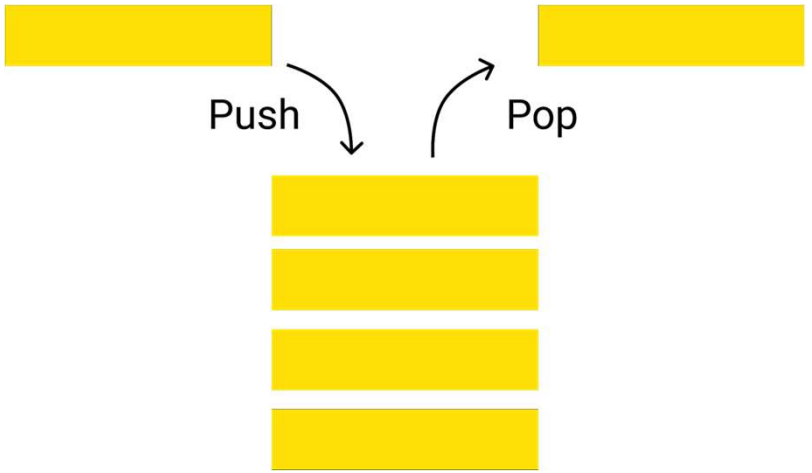
Познакомиться со структурами данных и алгоритмами:

- хэш-таблица, стек, очередь, куча
- List, Set
- Map/WeakMap
- Деревья
- Базовые алгоритмы
- Big O notation

Стек:

Стек следует принципу **LIFO (Last In First Out)**.

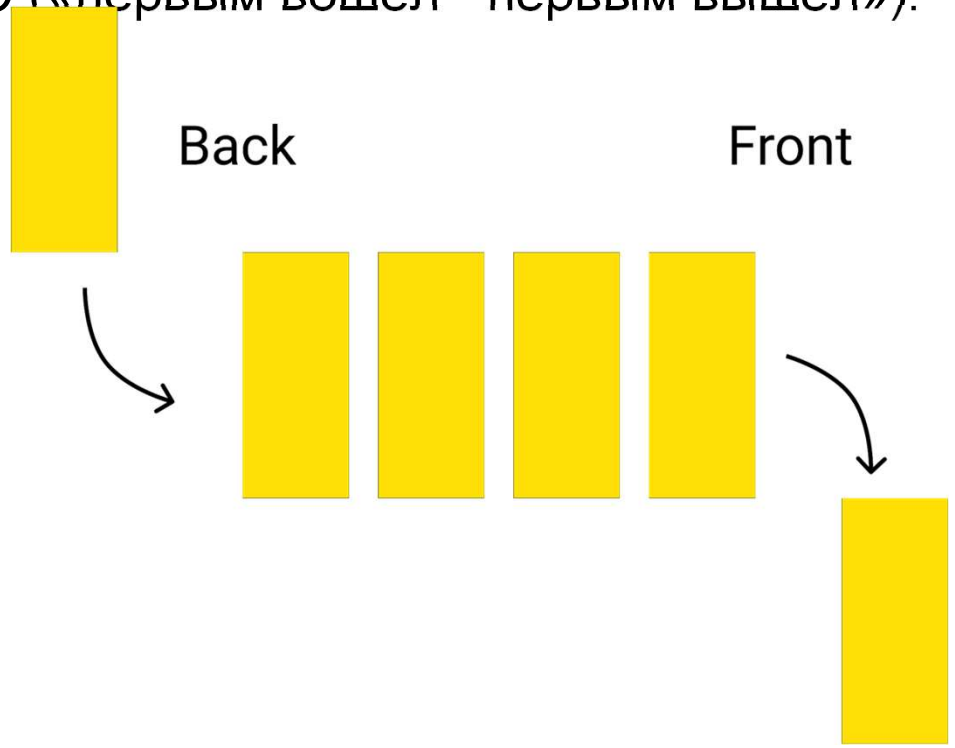
- **push**: введите новый элемент
- **pop**: удалить верхний элемент, вернуть удаленный элемент
- **peek**: вернуть верхний элемент
- **length**: вернуть количество элементов в стеке



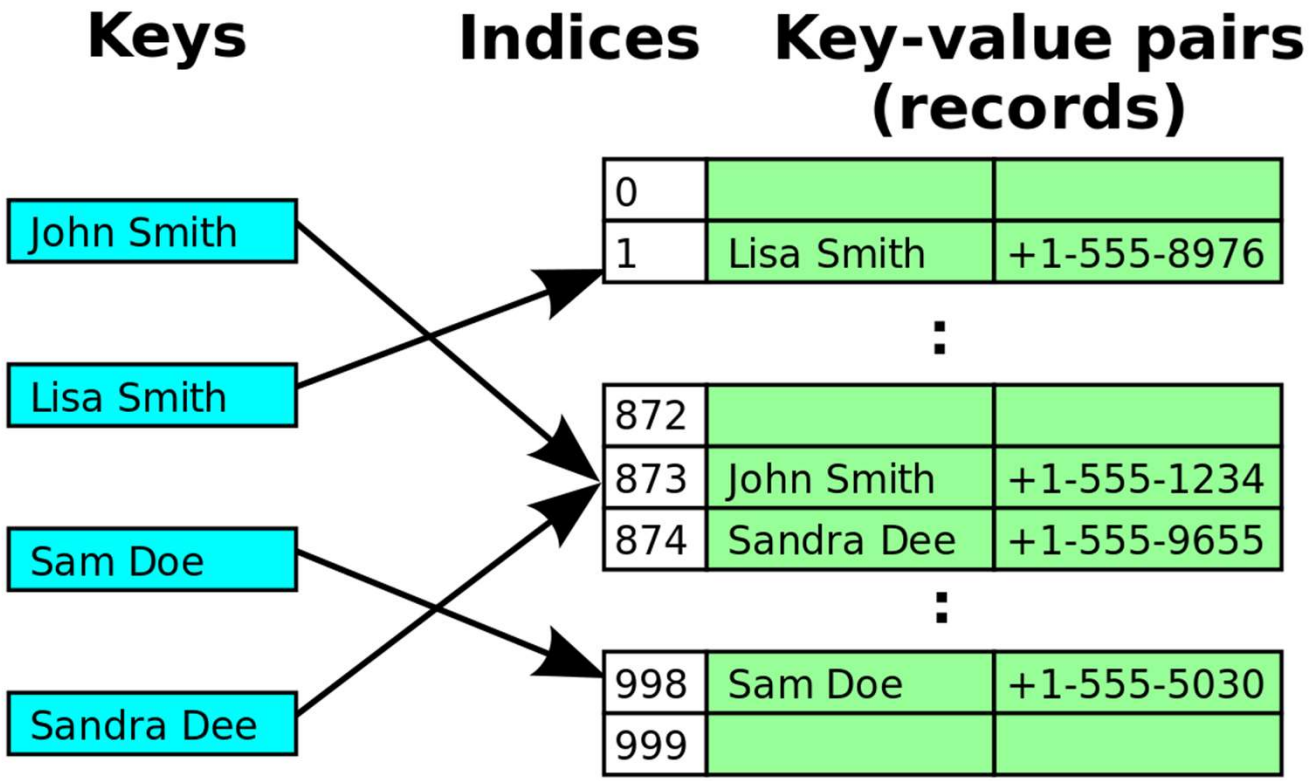
Очередь:

В очереди используется принцип FIFO («первым вошел - первым вышел»).

- **enqueue**: введите очередь, добавьте элемент в конце
- **dequeue**: покинуть очередь, убрать передний элемент и вернуть его
- **front**: получить первый элемент
- **isEmpty**: определить, пуста ли очередь
- **size**: получить количество элементов в очереди

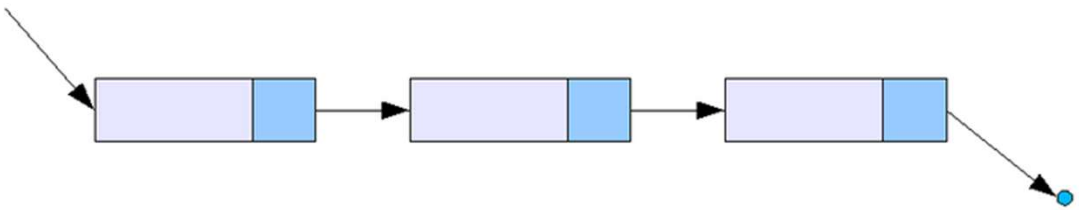


Хеш-таблица:


Keys
Indices
Key-value pairs (records)
John Smith
Lisa Smith
Sam Doe
Sandra Dee
0
1
Lisa Smith
+1-555-8976
872
873
John Smith
+1-555-1234
874
Sandra Dee
+1-555-9655
998
Sam Doe
+1-555-5030
999

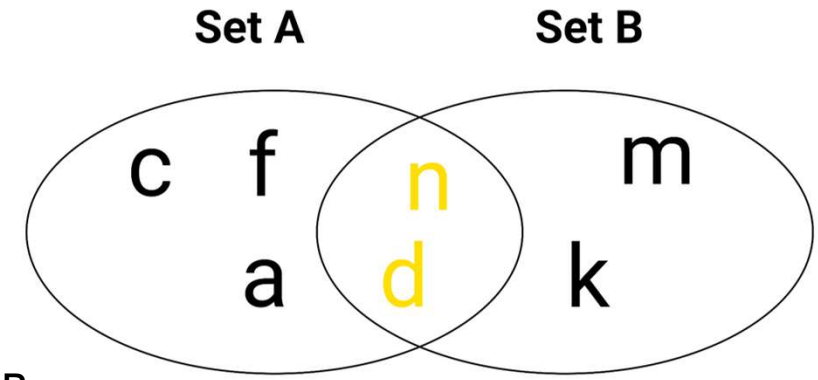
Связанный список:

Связанный список буквально представляет собой цепочечную структуру данных, где каждый узел состоит из двух частей информации: данных узла и указателя на следующий узел. Связанный список и обычный массив являются линейными структурами данных с сериализованным хранилищем.



Set:

- **values**: Вернуть все элементы в наборе
- **size**: Вернуть количество элементов
- **has**: Определить, существует ли элемент
- **add**: Вставить элементы в набор
- **remove**: Удалить элементы из набора
- **union**: Вернуть пересечение двух множеств
- **difference**: Вернуть разницу двух комплектов
- **subset**: Определить, является ли определенный набор подмножеством другого набора



Map:

Map – коллекция для хранения записей вида ключ:значение. В отличие от объектов, в которых ключами могут быть только строки, в Map ключом может быть произвольное значение.

Сохранения и чтения значений используются методы get и set. И ключи и значения сохраняются «как есть», без преобразований типов.

Map:

- new Map() – создает коллекцию.
- map.set(key, value) – записывает по ключу key значение value.
- map.get(key) – возвращает значение по ключу или undefined, если ключ key отсутствует.
- map.has(key) – возвращает true, если ключ key присутствует в коллекции, иначе false.
- map.delete(key) – удаляет элемент по ключу key.
- map.clear() – очищает коллекцию от всех элементов.
- map.size – возвращает текущее количество элементов.

Map:

- map.keys() – возвращает итерируемый объект для ключей,
- map.values() – возвращает итерируемый объект для значений,
- map.entries() – возвращает итерируемый объект для записей [ключ, значение], он используется по умолчанию в for..of.

WeakMap:

WeakMap – особый вид Map, не препятствующий сборщику мусора удалять свои элементы.

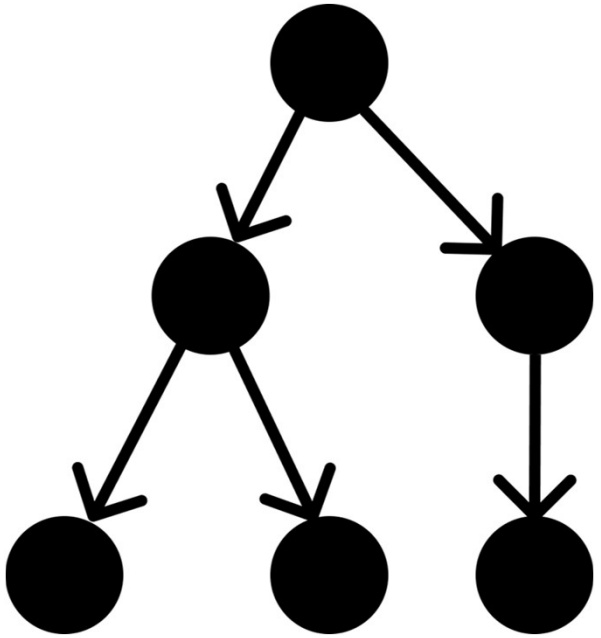
Это нужно для тех ситуаций, когда основное место для хранения и использования объектов находится где-то в другом месте кода, а здесь мы хотим хранить для них «вспомогательные» данные, существующие лишь пока жив объект.

У WeakMap есть ряд ограничений:

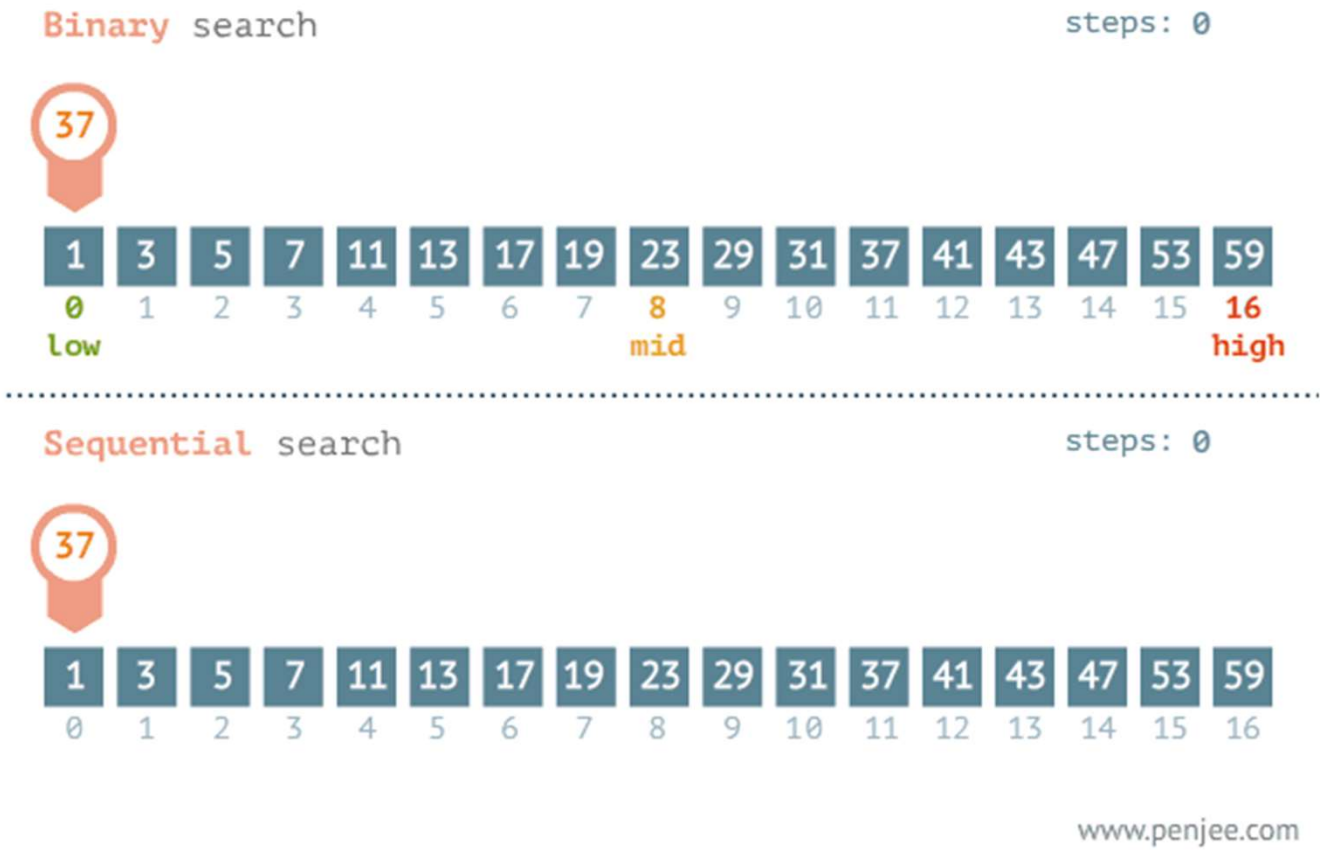
- Только объекты в качестве ключей.
- Нет свойства size.
- Нельзя перебрать элементы итератором или forEach.
- Нет метода clear().

Дерево:

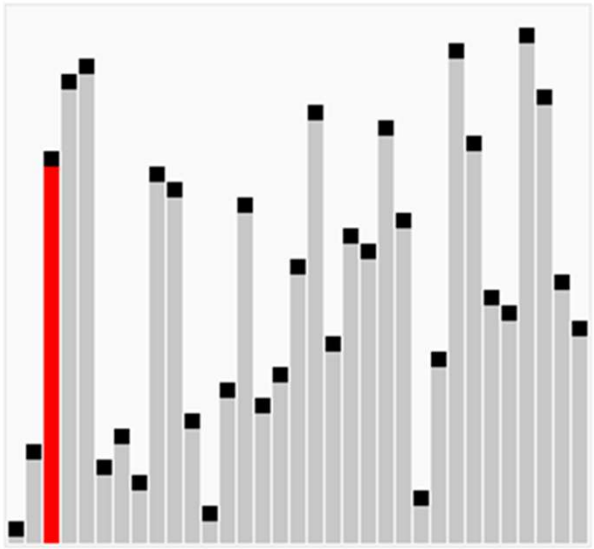
- root: Корневой узел дерева, нет родительского узла для корня
- parent node: Прямой узел верхнего слоя, только один
- child node: Прямой узел (узлы) нижнего уровня, может иметь несколько
- siblings: Общий родительский узел
- leaf: Узел без потомка
- Edge: Ветвь или связь между узлами
- Path: Ребра от начального узла до целевого узла
- Height of Node: Число ребер самого длинного пути конкретного узла к конечному узлу
- Height of Tree: Число ребер самого длинного пути корневого узла к листовому узлу
- Depth of Node: Количество ребер от корневого узла до определенного узла
- Degree of Node: Количество дочерних узлов



Линейный/Бинарный поиск:



Сортировка пузырьком:



Сортировка выбором:

Худший случай

Big O notation

Big O нотация нужна для описания сложности алгоритмов.