Структуры 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.

**Структуры данных делятся на 2 типа:**

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

**Нелинейные,** если обход узлов нелинейный, а данные не последовательны. Пример: граф, деревья.

**Стандартные операции для структур данных:**

1. **Создание** структуры данных.
2. **Вставка** данных в структуру. Есть три варианта:

* **В начало.**
* **В конец.**
* **В конкретное место.**

1. **Обход** каждого элемента как минимум один раз.
2. **Поиск** элемента среди других. Есть два вида:

* **Линейный** (последовательный)
* **Бинарный** (двоичный, полуинтервальный, логарифмический)

1. **Сортировка**  (Выстраивание элементов в нужном порядке)
2. **Слияние** (объединение элементов из двух структур)
3. **Обновление** (замена значения в структуре новым значением)
4. **Удаление** элемента из структуры.

Простые структуры данных

**Примитивы (примитивные типы) –** это базовые элементы в классификации структур данных.

**К ним относятся:**

1. **Логические(булевы)** – они могут иметь лишь два состояния истина или лож.
2. **Целочисленные** – содержат в себе целые числа.
3. **Числа с плавающей запятой –** содержат вещественные числа.
4. **Строковые** – содержат последовательность символов.
5. **Указатели** – переменная, которая состоит из адресов ячеек памяти.
6. **Идентификационные** – являются уникальным идентификатором объекта.
7. **Абстрактные** – рассматриваются независимо от контекста.

Сложные структуры данных

1. Массивы

**Массив** — это самая простая и широко используемая структура данных. Другие структуры данных, такие как стеки и очереди, являются производными от массивов. Каждому элементу данных присваивается положительное числовое значение (индекс), который соответствует позиции элемента в массиве. Большинство языков определяют начальный индекс массива как 0.

**Массивы делятся на:**

**Одномерный (вектор)** – каждый элемент лежит друг за другом в одну строку.

**Многомерный** – это массивы внутри массивов.

1. Строка

**Строки** определяются как массив символов. Разница между массивом символов и строкой заключается в том, что строка завершается специальным символом ‘\0’.

1. Связанный список

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

Принципиальным преимуществом перед массивом является структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями.

**Связные списки делятся на три вида:**

**Однонаправленный** - каждый узел хранит адрес или ссылку на следующий узел в списке и последний узел имеет следующий адрес или ссылку как NULL.

**Двунаправленный(Двусвязный)** - две ссылки, связанные с каждым узлом, одним из опорных пунктов на следующий узел и один к предыдущему узлу.

**Круговой** - все узлы соединяются, образуя круг. В конце нет NULL. Циклический связанный список может быть одно-или двукратным циклическим связанным списком.

Стеки

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

Примером стека может быть куча книг, расположенных в вертикальном порядке. Для того, чтобы получить книгу, которая где-то посередине, вам нужно будет удалить все книги, размещенные на ней. Так работает метод LIFO (Last In First Out). Функция «Отменить» в приложениях работает по LIFO.

Очереди

**Очередь** — подобна стеку, хранит элемент последовательным образом. Существенное отличие от стека – использование FIFO (First in First Out) вместо LIFO.

Пример очереди – очередь людей. Последний занял последним и будешь, а первый первым ее и покинет.

Хэш таблицы

**Хеширование** — 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.

Эффективность хеширования зависит от:

* Функции хеширования
* Размера хэш-таблицы
* Метода борьбы с коллизиями

Деревья

**Дерево** — это иерархическая структура данных, состоящая из узлов (вершин) и ребер (дуг). Деревья, по сути, связанные графы без циклов.

**Типы деревьев**

* N дерево
* Сбалансированное дерево
* Бинарное дерево
* Дерево Бинарного Поиска
* AVL дерево
* 2-3-4 деревья

Три способа обхода дерева

* В прямом порядке (сверху вниз) — префиксная форма.
* В симметричном порядке (слева направо) — инфиксная форма.
* В обратном порядке (снизу вверх) — постфиксная форма.

Графы

**Граф** — это набор узлов (вершин), которые соединены друг с другом в виде сети ребрами (дугами).

**Графы делятся на два вида**:

**Ориентированный** - ребра являются направленными, т.е. существует только одно доступное направление между двумя связными вершинами.

**Неориентированные** - к каждому из ребер можно осуществлять переход в обоих направлениях.

**Встречаются в таких формах как:**

* Матрица смежности
* Список смежности

**Общие алгоритмы обхода графа:**

* Поиск в ширину – обход по уровням
* Поиск в глубину – обход по вершинам