**DATA STRUCTURES AND ALGORITHMS.**

**HYBRID APPROACH (DRAKON+GOLANG)**

Edition 3.2 (C++ Version) Clifford A. Shaffer Department of Computer Science Virginia Tech Blacksburg, VA 24061 January 2, 2012 Update 3.2.0.3 For a list of changes, see http://people.cs.vt.edu/˜shaffer/Book/errata.html

**2022**

ВВЕДЕНИЕ

Основу процессов познания, понимания и конструирования природной, неприродной и виртуальной реальностей так или иначе составляют ДАННЫЕ, которые можно понимать как неструктурированную информацию. Для структуризации этих данных с целью получения нового знания или новых смыслов необходима их обработка по определенному плану действий. В более строгом определении структура данных – это форма представления свойств и связей предметной области, ориентированная на выражение описания данных средствами формальных язы­ков. Технологически структура данных — это контейнер, информация в котором скомпонована определенным образом, устанавливающим способ размещения данных в памяти компьютера для быстрого и эффективного доступа к ним.

В свою очередь алгоритмы представляют собой наборы инструкций для решения вычислительных проблем путем обработки структур данных. Структуры данных и процессы их обработки по тем или иным алгоритмам неразрывно связаны друг с другом.  В одних задачах выбор алгоритма диктуется структурой данных, в других – структура данных определяет выбор алгоритма. Один из основоположников становления информатики как академической науки Н. Вирт сформулировал эту связь в названии книги: «Алгоритмы + структуры данных = программа» [Вирт].

Многообразие предметного, непредметного и виртуального миров и процессов, происходящих в них, предопределяют возникновение множества проблемных ситуаций, эффективное решение которых которые предполагает выбор структур данных и алгоритмов их обработки. Решение считается [эффективным](https://opendsa-server.cs.vt.edu/ODSA/Books/Everything/html/Glossary.html#_blank) , если оно решает проблему в рамках известных  [ограничений ресурсов](https://opendsa-server.cs.vt.edu/ODSA/Books/Everything/html/Glossary.html#_blank). Прежде всего имеется в виду *общее пространство*, доступное для хранения данных, а также *время*, отведенное на выполнение каждого алгоритма. При выборе структуры данных для решения задачи следует, во-первых, определить основные операции, которые должны поддерживаться, во-вторых, количественно оценить ограничения ресурсов по каждой операции алгоритма (занимаемой памяти и времени выполнения операций).

Структура данных требует определенного объема памяти для каждого хранимого элемента данных, определенного количества времени для выполнения одной базовой операции и определенных усилий по программированию. Каждая задача имеет ограничения по доступному пространству и времени. Каждое решение проблемы использует базовые операции в некоторой относительной пропорции, и процесс выбора структуры данных должен это учитывать. Только после тщательного анализа характеристик вашей задачи вы сможете определить наилучшую структуру данных для этой задачи [].

Исходя из представленных соображений в Пособии даются базовые сведения о наиболее часто используемых структурах данных и алгоритмов, поскольку они составляют основу профессиональной деятельности программиста любого уровня. Уделяется значительное внимание вопросам оценки эффективности структур данных и алгоритмов с точки зрения занимаемой памяти и времени выполнения базовых операций. Организация структур данных и изучение базовых алгоритмов выполняется на языке программирования Golang и сопровождается поясняющими иллюстрациями. В ряде важных случаях применяется технология визуализация алгоритмов на основе графического языка Drakon.

В **РАЗДЕЛЕ 1** раскрывается смысл основных понятий в дисциплине *Структура данных и алгоритмы*: данные, тип данных, абстрактные типы данных, алгоритмы. Высокий уровень компетентности в сфере разработки эффективных алгоритмов необходим каждому программисту не только при решении практических проблем, связных с применением IT-технологий. Владение знаниями в области структуры данных и алгоритмов является важнейшей составляющей при собеседовании при приему на работу в крупные IT-компании.

Во **ВТОРОМ РАЗДЕЛЕ** излагаютсяосновы языка программирования Golang в объеме, достаточном для знакомства и практического овладения знаниями и навыками в области структуры данных и алгоритмов. Разумеется, для прочтения этой главы предполагается наличие базовых понятий и определенный уровень программирования.

В **РАЗДЕЛЕ 3** дается краткое описание визуального языка DRAKON, позволяющего представлять сколь-угодно сложные алгоритмы в виде дракон-диаграммы с помощью графоэлементов, соответствующих основным конструкциям языка Golang. Приводится технология автоматизированного преобразования дракон-диаграммы в программный код в редакторе DRAKON WEB Editor с последующей реализацией в оболочке Visual Stufio Code. Собственно , в этом и заключается гибридный, двухэтапный подход: на первом этапе составляется дракон-диаграмма, гоафоэлементы которой заполняются конструкциями языка Golang, на втором — автоматически генерируется программны код, реализуемый в какой-либо программной средею

**В РАЗДЕЛЕ 4** представлены конструкции языка Golang, реализующие основные абстрактные типы данных, описанные в Разделе 1: массив, срез, спискии т. д. Приводятся пояснения в отношении использования соответствующих конструкций языка Golang.

В **РАЗДЕЛЕ** **5** изложены основные представления об анализе сложности алгоритмов и об оценке и эффективности алгоритмов с позиций использования компьютерной памяти (пространство) и затрат время. Теоретические представления об оценке сложности алгоритмов имеют определяющее значение при выборе наиболее эффективного из них.

**РАЗДЕЛ 6** посвящен базовым алгоритмам сортировки данных. Каждый алгоритм представлен с помощью соответствующей дракон-диаграммы, сопровождающейся необходимыми пояснениями о процессе обработки и оценкой сложности.

В **РАЗДЕЛЕ 7** рассматриваются базовые алгоритмы поиска элементов в различных структурах данных (срезы, списки). Кроме того раздел включает описание и дракон-диаграммы алгоритмов хэширования, позволяющие значительно сократить время поиска за счет уменьшения количества сравнений значений элементов.

**РАЗДЕЛ 8** содержит понятийную информацию о двоичных деревьях - нелинейных структурах данных, включая двоичные деревья поиска и самобалансирубщиеся деревья. В разделе рассматриваются базовые ДРАКОН-диаграммы алгоритмов вставки новых, поиска узлов с указанными значениями и удаление узлов.

**В РАЗДЕЛЕ 9** излагаются базовые сведения о наиболее сложных структурах данных — графах. Приводятся основные понятия теории графов и основные их виды. Рассматриваются алгоритмы представления графов, их обхода, выбора пути между вершинами.