Тема 1

1. Алгоритм — это последовательность команд, предназначенная исполнителю, в результате выполнения которой он должен решить поставленную задачу. Алгоритм должен описываться на формальном языке, исключающем неоднозначность толкования.

Свойства:

* детерминированность (определенность). Предполагает получение однозначного результата вычислительного процecca при заданных исходных данных. Благодаря этому свойству процесс выполнения алгоритма носит механический характер;
* результативность. Указывает на наличие таких исходных данных, для которых реализуемый по заданному алгоритму вычислительный процесс должен через конечное число шагов остановиться и выдать искомый результат;
* массовость. Это свойство предполагает, что алгоритм должен быть пригоден для решения всех задач данного типа;
* дискретность. Означает расчлененность определяемого алгоритмом вычислительного процесса на отдельные этапы, возможность выполнения которых исполнителем (компьютером) не вызывает сомнений;
* конечность. Каждое из действий и весь алгоритм в целом обязательно завершаются.
* Понятность
* эффективность

1. Корректность алгоритма - Разработанный алгоритм, для всех исходных данных, для которых его задача может быть решена, позволяет получить правильный результат и ни для каких исходных данных не дает неправильного результата
2. Концепция типов данных:

Основная цель любой программы состоит в обработке данных. Данные различного типа хранятся и обрабатываются по-разному. В любом алгебраическом языке каждая константа, переменная, результат вычисления выражения или функции должны иметь определенный тип.

Тип данных определяет:

* Внешний уровень (Как видит данные пользователь будущей программной системы);
* Логический (Как реализует программист данные задачи);
* Физический (Характеризует данные в виде, воспринимаемом операционной системой, т.е. физическое хранение данных).

Исходя из этих характеристик, программист выбирает тип каждой величины, используемой в программе для представления реальных объектов. Обязательное описание типа позволяет компилятору производить проверку допустимости различных конструкций программы. От типа величины зависят машинные команды, которые будут использоваться для обработки данных. Все типы языка С++ можно разделить на основные и составные. В языке С++ определено шесть основных типов данных для представления целых, вещественных, символьных и логических величин. К ним относятся массивы, перечисления, функции, структуры, ссылки, указатели, объединения и классы.

1. Эффективность алгоритма зависит от множества факторов, которые включают:

• вычислительную систему – системная эффективность (быстродействие системы)

• объем доступной ОП вычислительной системы

• вычислительную сложность алгоритма

1. Семантика — это соответствие между синтаксически правильными программами и действиями абстрактного исполнителя, то есть это смысл синтаксических конструкций.

Прагматика–правила преобразования элементов программы, написанной на языке программирования, в элементы программы на другом языке, например в команды компьютера.

1. Структуры данных - это совокупность элементов данных и отношений между ними. При этом под элементами данных может подразумеваться как простое данное так и структура данных. Под отношениями между данными понимают функциональные связи между ними и указатели на то, где находятся эти данные.
2. Линейные структуры — это упорядоченные структуры, в которых адрес элемента однозначно определяется его номером. (Массивы, Связанный список, стеки и очереди)
3. Структуры данных классифицируются:

1. По связанности данных в структуре:

- если данные в структуре связаны очень слабо, то такие структуры называются несвязанными (вектор, массив, строки, стеки)

- если данные в структуре связаны, то такие структуры называются связанными (связанные списки)

2. По изменчивости структуры во времени или в процессе выполнения программы:

- статические структуры - структуры, неменяющиеся до конца выполнения программы (записи, массивы, строки, вектора)

- полустатические структуры (стеки, деки, очереди)

- динамические структуры - происходит полное изменение при выполнении программы

3. По упорядоченности структуры:

- линейные (вектора, массивы, стеки, деки, записи)

- нелинейные (многосвязные списки, древовидные структуры, графы)

Наиболее важной характеристикой является изменчивость структуры во времени.

1. Свойства структур:

* Линейная или нелинейная структура данных
* Однородная или неоднородная структура данных
* Статическая или динамическая структура
* Способ доступа к элементу структуры данных

1. Существуют следующие способы представления алгоритмов:

* словесное описание;
* описание алгоритма с помощью математических формул;
* графическое представление алгоритма в виде блок-схемы;
* представление алгоритма с помощью псевдокода;
* комбинированный способ описания алгоритма с использованием, например, словесного и графического способов или словесного и с помощью математических формул и т.д.

1. Модель данных – это совокупность взаимосвязанных структур данных и операций над этими структурами.

Тема 2

1. О - позволяет сравнивать количество необходимых операций. Она говорит о том, как быстро возрастает скорость работы алгоритма. Эта запись сообщает количество операций, которое осуществит алгоритм. Ограничена сверху, точная оценка неизвестна

Θ – ограничена сверху и снизу, точная оценка

Ω - ограничена снизу, точная оценка неизвестна

o – бесконечно малая функция, ограничена сверху, неточная оценка

ω – ограничена снизу, неточная оценка

1. Свойства асимптотических функций:

* Транзитивность
* Рефлексивность
* Симметричность
* Обращение

1. Способы оценки вычислительной сложности алгоритмов:

* Подсчет количества выполняемых операций

1. Оценки эффективности в наилучшем, наихудшем, среднем случаях и их применение при выборе оптимального алгоритма:
2. Рекурсия – это функция, которая вызывает саму себя. Рекурсия, при которой рекурсивные вызовы на любом рекурсивном срезе, инициируют не более одного последующего рекурсивного вызова, называется линейной. Любая рекурсия, которая делает более одного вызова себя, называется каскадной. Рекурсия называется удаленной, если в теле функции при рекурсивных вызовах, в выражениях, являющихся фактическими параметрами, снова встречаются рекурсивные вызовы этой функции

Шаг рекурсии - это правило, в теле которого обязательно содержится, в качестве подцели, вызов определяемого предиката. n – 1 - это шаг рекурсии. При последующем вызове функции мы передаем ей число на единицу большее, чем получили. Это выполнение одного действия

Объем рекурсии – это количество вершин полного рекурсивного дерева без единицы.

Дерево рекурсии – это полное дерево, т.е. на каждом уровне, кроме последнего располагается максимально допустимое количество узлов.

Удаленная рекурсия - если в определении рекурсивной функции вызов функции в списке фактических параметров содержит вызов самой функции.

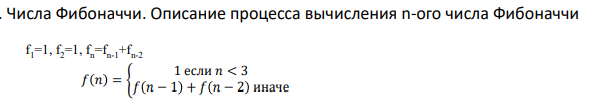
Глубина рекурсивных вызовов – наибольшее одновременное количество рекурсивных вызовов функции, определяющее максимальное количество слоев рекурсивного.

стека, в котором осуществляется хранение отложенных вычислений

Глубина рекурсии — это глубина дерева рекурсивных вызовов, то есть длина максимального пути по стрелочкам из вершины до одного из элементарных (базовых) значений функции.

Условие выхода из рекурсии - определяет завершение рекурсии и формирование конкретного значения вычислительного процесса.

Рекуррентное соотношение — это уравнение или неравенство, описывающее функцию с использованием ее самой.

1. 
2. <https://habr.com/ru/post/188010/>
3. Линейный список — это динамическая структура данных, каждый элемент которой посредством указателя связывается со следующим элементом.
4. Стек — это структура данных типа LIFO (англ. «Last In, First Out» = «Последним пришел, первым ушел»). Последний элемент, который находится на вершине стека, первым и уйдет из него. Если положить новую тарелку поверх других тарелок, то именно эту тарелку вы первой и возьмете.
5. Очередь — это структура данных (как было сказано выше), которая построена по принципу LILO (last in — last out: последним пришел — последним вышел). В C++ уже есть готовый STL контейнер — queue. В очереди, если вы добавите элемент, который вошел самый первый, то он выйдет тоже самым первым.
6. Дек, deque (двусторонняя очередь, очередь с двумя концами) — абстрактный тип данных, в котором элементы можно добавлять и удалять как в начало, так и в конец.
7. Постфиксная (или обратная польская) запись арифметического выражения — это способ записи выражений, в котором знак операции записывается не между операндами, а после операндов.