Определение алгоритма по Маркову

**Алгоритм** – это точное предписание, **определяющее** вычислительный процесс, идущий от варьируемых исходных данных к искомому результату.

Определение алгоритма по Колмагорову

**Алгоритм** – это всякая система вычислений, выполняемых по строго **определенным** правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи.

Понятие о трудоемкости алгоритма

**Трудоемкость алгоритма – это зависимость количества массовых операций (сравнения, обмены, сдвиги, повторения цикла и т.п.) от объема обрабатываемых данных.**

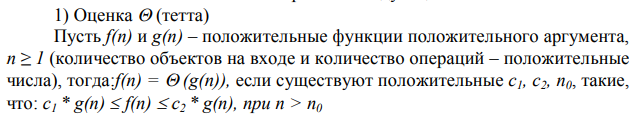
Самое важное, что трудоемкость напрямую не связана со временем выполнения программы, но является мерой затрат на ее выполнение.

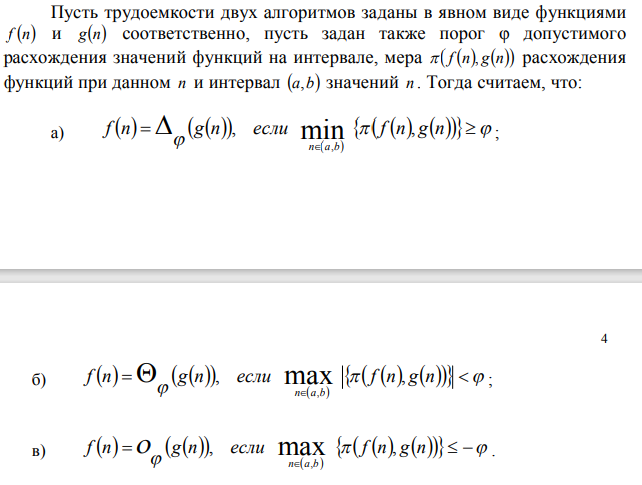
Асимптотические обозначения в анализе функции трудоемкости алгоритма.

Основные оценки в асимптотическом анализе алгоритмов

**Запись вида f(n) = O(g(n)) означает, что ф-ия f(n) возрастает медленнее чем ф-ия g(n) при с = с1 и n = N, где c1 и N могут быть сколь угодно большими числами, т.е. При c = c1 и n >= N, c\*g(n) >=f(n).**

Таким образом O – означает верхнее ограничение сложности алгоритма.





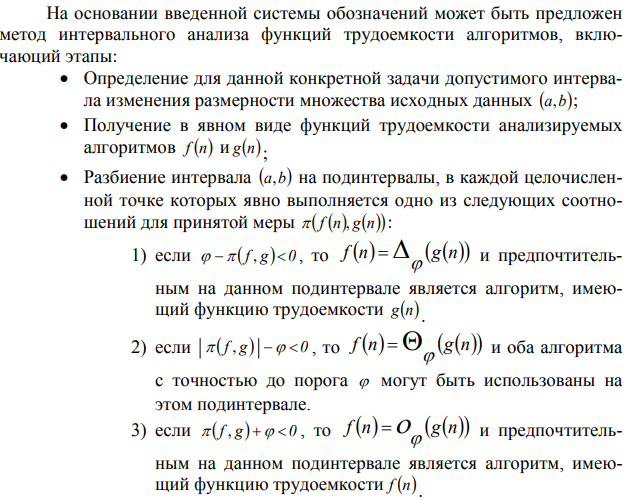
**Определение сложности алгоритма**

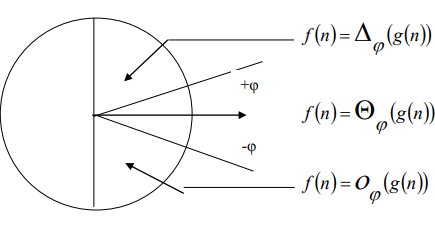
**сложность алгоритма-это количество ресурсов, необходимых для его запуска**. Особое внимание уделяется требованиям ко времени и памяти.

Классификация алгоритмов по трудоёмкости О().

О (1), О (n), О (n2), О (n3), О (log n), О (n\*log n), О (2n)

Метод интервального анализа функций трудоемкости алгоритмов, его графическая интерпретация





ЛР2

Определение детерминированного и рандомизированного алгоритмов

Детерминированный алгоритм задаёт определённые действия обозначая их в единственной и достоверной последовательности.

Рандомизированный алгоритм это **алгоритм**, который использует определенную степень случайности как часть своей логики или процедуры

Определение генератора псеводслучайных чисел

**Генератор** **псевдослучайных** **чисел** (**ГПСЧ**, англ. pseudorandom number generator, PRNG) — алгоритм, порождающий последовательность **чисел**, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно дискретному равномерному).

Описать основные функции и константы, использующиеся для создания последовательности псевдослучайных чисел в языке С++

#include <ctime>

void create\_array (int n) {

srand (time (0));

for (int i=1; i<=n; i++)

vector[i-1] = rand() % 950;

}

ЛР 3

Обозначения лучшего, худшего, среднего случая в анализе трудоемкости алгоритма

Под худшим случаем трудоемкости понимают наибольшее количество операций, задаваемых алгоритмом

лучший случай трудоемкости, как наименьшее количество операций в аналогичном алгоритме и при той же размерности входа

Средний случай трудоемкости определяется средним количеством операций рассматриваемого алгоритма и входных данных

Классификация алгоритмов по виду функции трудоемокости

*Количественно-зависимые по трудоемкости алгоритмы*

Это алгоритмы, функция трудоемкости которых зависит только от размерности конкретного входа, и не зависит от конкретных значений

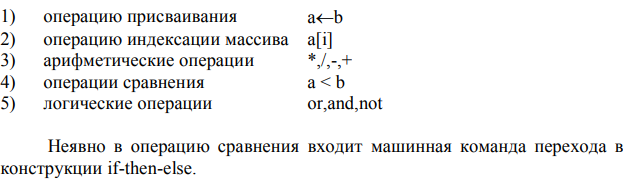
*Параметрически-зависимые по трудоемкости алгоритмы*

Это алгоритмы, трудоемкость которых определяется не размерностью входа (как правило, для этой группы размерность входа обычно фиксирована), а конкретными значениями обрабатываемых слов памяти

*Количественно-параметрические по трудоемкости алгоритмы*

Однако в большинстве практических случаев функция трудоемкости зависит как от количества данных на входе, так и от значений входных данных

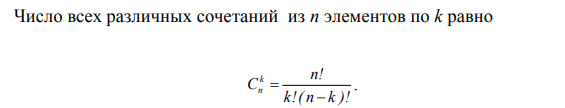
Перечень элементарных операций языка высокого уровня



Формулировка задачи о сумме

Словесно **задача** **о** **сумме** формулируется как **задача** нахождения таких чисел из данной совокупности, которые в **сумме** дают заданное число. В терминах языка высокого уровня **задача** формулируется как **задача** определения таких элементов исходного массива из N чисел, которые в **сум**-**ме** дают число V (**задача** относится к классу NPC).

Основные комбинаторные формулы, использующиеся в задаче о сумме



ЛР 5

Дайте определение пути, маршрута, цепи, контура

Путем называется такая последователь­ность дуг, когда конец каждой предыдущей дуги совпадает с началом последующей. Понятие пути обычно ис­пользуется для ориентированных графов.

Цепью называется та­кая последовательность ребер *Е0, E1, ..., Ek-1, Ek,*  
когда каждое ребро *Еk-1* соприкасается одним из концов с ребром *Еk.*

Понятие цепи обычно используется для неориентированных графов.

Цепь можно обозначить последовательностью вершин, которые она содержит. Например, для графа це­пью будет (1, 4, 3, 5) или (1, 3, 4).

Маршрут есть неориентированный "двойник" пути, и это понятие рассматривается в тех случаях, когда можно пренебречь направленностью дуг в графе.

Контуром (простой цепью) называется такой путь (маршрут), в котором каждая вершина используется не более одного раза.

Ориентированной цепью (орцепью) называется такой путь, в котором каждая дуга используется не больше одного раза.

Какой граф называется взвешенным

это граф (орграф)**, некоторым элементам которого (вершинам, ребрам или дугам) сопоставлены числа**.

Как определяется длина пути графа

Длина графа это сумма длин (весов) тех рёбер из которых состоит путь.

Задача нахождения кратчайшего пути на графе

В теории **графов** **задача** **кратчайшего** **пути**-это **задача** **нахождения** **пути** между двумя вершинами (или узлами) **графа** таким образом, чтобы сумма весов составляющих его ребер была минимизирована. **Задача** **нахождения** **кратчайшего** **пути** между двумя пересечениями на дорожной карте может быть смоделирована как частный случай **задачи** **кратчайшего** **пути** в **графах**, где вершины соответствуют пересечениям, а ребра-отрезкам дороги, каждый из которых взвешен по длине отрезка.

Реализация метода динамического программирования для нахождения кратчайшего пути на графе

**ЛР №3**

**“Экспериментальное определение количества элементарных операций языка**

**высокого уровня в программной реализации алгоритма”**

**1. Определение трудоёмокости алгоритма.**

Трудоемкость алгоритма – это зависимость количества массовых операций (сравнения, обмены, сдвиги, повторения цикла и т.п.) от объема обрабатываемых данных.

Самое важное, что трудоемкость напрямую не связана со временем выполнения программы, но является мерой затрат на ее выполнение.

**2. Обозначения лучшего, худшего, среднего случая в анализе трудоемкости алгоритма.**

**3. Классификация алгоритмов по виду функции трудоемокости.**

**4. Перечень элементарных операций языка высокого уровня.**

**5. Формулировка задачи о сумме.**

**6. Основные комбинаторные формулы, использующиеся в задаче о сумме.**