

Википедия

Список алгоритмов

Материал из Википедии — свободной энциклопедии

Ниже приводится **список алгоритмов**, группированный по категориям. Более детальные сведения приводятся в списке структур данных и [списке основных разделов теории алгоритмов](#)^[1]

Содержание

Комбинаторные алгоритмы

- Общие комбинаторные алгоритмы
 - Генерация комбинаторных объектов
- Алгоритмы на графах
 - Алгоритмы нахождения максимального потока
 - Алгоритмы нахождения максимального паросочетания
- Алгоритмы поиска
- Алгоритмы на строках
 - Алгоритмы поиска строки
 - Алгоритмы вычисления расстояния между строками
 - Алгоритмы приближенного сравнения строк с шаблоном
 - Вычисление характеристических паттернов
 - Примерное соответствие
 - Деревья для строковых последовательностей
- Алгоритмы сортировки
- Алгоритмы слияния
- Минимизация булевых функций

Алгоритмы сжатия данных

- Алгоритмы сжатия без потерь
- Алгоритмы сжатия с потерями

Вычислительная геометрия

- Построение выпуклой оболочки набора точек
- Триангуляция
 - Триангуляция Делоне
- Квазитриангуляция
- Диаграмма Вороного
- Локализация точки (англ.)
- Пересечения
- Вращающиеся калиперы (англ.)

Компьютерная графика

Компьютерное зрение

Криптографические алгоритмы**Цифровая обработка сигналов****Разработка программного обеспечения**

- Алгоритмы распределённых систем
- Алгоритмы выделения и освобождения памяти
- Алгоритмы в операционных системах
- Дисковые алгоритмы-планировщики
- Сетевые алгоритмы
- Алгоритмы синхронизации процессов
- Алгоритмы планирования

Генетические алгоритмы**Медицинские алгоритмы****Нейронные сети****Вычислительная алгебра****Теоретико-числовые алгоритмы****Численные алгоритмы****Алгоритмы оптимизации****Грамматический разбор****Квантовые алгоритмы****Теория вычислений и автоматов****Другие****См. также****Примечания****Литература****Ссылки**

Комбинаторные алгоритмы

Общие комбинаторные алгоритмы

- Алгоритм Флойда для нахождения циклов находит цикл в итерациях
- Генераторы псевдослучайных чисел:
 - Алгоритм Блум — Блюма — Шуба
 - Вихрь Мерсенна
 - Метод Фибоначчи с запаздываниями
 - Линейный конгруэнтный метод
 - Инверсный конгруэнтный метод

Генерация комбинаторных объектов

- Алгоритм Робинсона — Шенстеда — генерация перестановок из пар таблиц Юнга

- [Алгоритм Нарайаны](#)
- [Алгоритм Джонсона — Троттера](#)

Алгоритмы на графах

- [Алгоритм Беллмана — Форда](#) — вычисляет кратчайший путь во взвешенном графе (где некоторые веса рёбер могут быть отрицательны)
- [Алгоритм Борушки](#) — находит минимальное остовное дерево в графе
- [Алгоритм Брона — Кербоша](#) — нахождение наибольших максимальных независимых по включению множеств вершин графа.
- [Алгоритм Флойда — Уоршелла](#) — вычисляет все кратчайшие пути во взвешенном графе
- [Алгоритм Дейкстры](#) — вычисляет кратчайший путь в графе с неотрицательными весами рёбер
- [Алгоритм Левита](#) — находит кратчайшее расстояние от одной из вершин графа до всех остальных
- [Алгоритм Джонсона](#) — вычисляет все кратчайшие пути во взвешенном графе
- [Алгоритм Краскала](#) — находит остовный лес минимального веса в графе
- [Алгоритм, основанный на источнике](#) — алгоритм для рисования графа
- [Алгоритм Прима](#) — находит остовное дерево минимального веса в связном графе
- [Алгоритм Кристофидеса](#) — эвристический приближенный алгоритм для решения метрической задачи коммивояжера на графе.
- [Метод ближайшего соседа](#) — «жадный» алгоритм, один из простейших эвристических методов решения задачи коммивояжёра
- [Неблокирующий минимальный охватывающий переключатель](#) например, для телефонной связи
- Построение транзитивного замыкания графа (установление факта достижимости вершин)

Алгоритмы нахождения максимального потока

n — число вершин, m — число рёбер, U — наибольшая величина максимальной пропускной способности сети.

- [Алгоритм Форда — Фалкерсона](#) (1956) — $O(nmU)$.
- [Алгоритм Эдмондса — Карпа, кратчайших увеличивающихся цепей](#) (1969) — $O(nm^2)$.
- [Алгоритм Диница](#) (1970) — $O(n^2m)$.
- [Алгоритм Эдмондса — Карпа, локально-максимального увеличения](#) (1972) — $O(m^2 \log U)$.
- [Алгоритм Диница 2](#) (1973) — $O(nm \log U)$.
- [Алгоритм Карзанова](#) (1974) — $O(n^3)$.
- [Алгоритм Черкасского](#) (1977) — $O(n^2 \sqrt{m})$.
- [Алгоритм Малхотры — Кумара — Махешвари](#) (1977) — $O(n^3)$.
- [Алгоритм Галила](#) (1980) — $O(n^{5/3} m^{2/3})$.
- [Алгоритм Галила — Наамада](#) (1980) — $O(nm \log^2 n)$.
- [Алгоритм Слейтора — Тарьяна](#) (1983) — $O(nm \log n)$.
- [Алгоритм Габоу](#) (1985) — $O(nm \log U)$.
- [Алгоритм Голдберга — Тарьяна](#) (1988) — $O(nm \log(n^2/m))$.

- Алгоритм Ахьюа — Орлина (1989) — $O(nm + n^2 \log U)$.
- Алгоритм Ахьюа — Орлина — Тарьяна (1989) — $O(nm \log(n\sqrt{U}/(m+2)))$.
- Алгоритм Кинга — Рао — Тарьяна 1 (1992) — $O(nm + n^{2+\epsilon})$.
- Алгоритм Кинга — Рао — Тарьяна 2 (1994) — $O(nm \log_{m/n} n)$.
- Алгоритм Черияна — Хейджрапа — Мехлхорна (1996) — $O(n^3 / \log n)$.
- Алгоритм Голдберга — Рао (1998) — $O(\min\{n^{2/3}, m^{1/2}\} m \log(n^2/m) \log U)$.
- Алгоритм Кёлнера — Мондры — Спилмана — Тена (2010) — $O(nm^{1/3} \epsilon^{-11/3} \log^c(nm^{1/3} \epsilon^{-11/3}))$.
- Алгоритм Орлина 1 (2012) — $O(nm)$.
- Алгоритм Орлина 2 (2012) — $O(n^2 / \log n)$, если $m = O(n)$.

Алгоритмы нахождения максимального паросочетания

- Алгоритм Хопкрофта — Карпа
- Алгоритм Форда — Фалкерсона
- Алгоритм Куна
- Алгоритм Габоу

Алгоритмы поиска

- Алгоритм поиска A^* — особый случай поиска по первому наилучшему совпадению; используется эвристика, увеличивающая скорость работы алгоритма
- Алгоритм выбора — модификация алгоритма линейного поиска; находит k -й по величине элемент в списке;
- Двоичное дерево поиска $O(\log n)$, $O(n)$ в худшем случае — использует бинарное дерево для хранения элементов;
 - Красно-чёрное дерево $O(\log n)$ — использует дополнительный атрибут узла дерева — «цвет»
 - АВЛ-дерево $O(\log n)$ — в каждом узле хранит разницу высот (целое число от -1 до $+1$)
 - Расширяющееся дерево $O(\log n)$ — вместо дополнительных полей в узлах дерева «расширяющие операции» выполняются при каждом обращении к дереву.
- Двоичный поиск $O(\log n)$ — находит элемент в отсортированном списке
- Интерполяционный поиск (Предсказывающий поиск, Поиск по словарю)
- Линейный поиск $O(n)$ — находит элемент в неотсортированном списке
- Локальный поиск (оптимизация)
- Метод штрафов
- Поиск в глубину — проходит граф ветка за веткой
- Поиск в ширину — проходит граф уровень за уровнем
- Поиск по первому наилучшему совпадению (англ. *Best-first search*) — проходит граф в порядке важности, используя очередь приоритетов
- Троичный поиск — находит максимум или минимум функции
- Поиск в хеш-таблице
- Алгоритм Ли (волновой алгоритм) — поиск пути на карте.

Алгоритмы на строках

Алгоритмы поиска строки

- [Префикс-функция](#)
- [Z-функция](#)
- [Алгоритм Кнута — Морриса — Пратта](#)
- [Алгоритм Рабина — Карпа](#) поиска строки
- [Алгоритм Бойера — Мура](#) поиска строки
 - [Алгоритм Бойера — Мура — Хорспула](#)
 - [Первый алгоритм Бойера — Мура — Санди](#)
 - [Второй алгоритм Бойера — Мура — Санди](#)
 - [Алгоритм Бойера — Мура — Гелила](#)
 - [Алгоритм Турбо-БМ](#)
- [Алгоритм Ахо — Корасик](#)
- [Алгоритм Битапа](#) (также известен как shift-or, shift-and)
- [Задача поиска наибольшей общей подпоследовательности](#)
- [Задача поиска наибольшей увеличивающейся подпоследовательности](#)
- [Задача поиска наикратчайшей общей надпоследовательности](#)
- [Задача поиска наибольшей общей подстроки](#)
- [Задача поиска количества подпалиндромов](#)

Алгоритмы вычисления расстояния между строками

- [Алгоритм Вагнера — Фишера](#)
- [Алгоритм Хешберга](#)
- [Алгоритм Ханта — Шиманского](#)
- [Алгоритм Укконена — Майерса](#)

Алгоритмы приближенного сравнения строк с шаблоном

- [Алгоритм Укконена](#)
- [Алгоритм Майерса](#)
- [Алгоритм Ву — Менбера](#)

Вычисление характеристических паттернов

- [Алгоритм Крочемора поиска всех кратных строк](#)
- [Алгоритм Мейна — Лоренца поиска всех кратных строк](#)
- [Алгоритм Мейна поиска крайних левых серий](#)
- [Алгоритм Колпакова — Кучерова поиска всех серий](#)
- [Алгоритм Ли — Смита поиска всех оболочек](#)
- [Алгоритм Франека — Смита — Танга поиска всех раппортов](#)
- [Алгоритм Шмидта поиска \$k\$ -приближенных раппортов](#)
- [Алгоритмы Сима — Илиопулоса — Парка — Смита поиска \$k\$ -приближенных периодов](#)

Примерное соответствие

- Расстояние Левенштейна
- Расстояние Хэмминга
- Расстояние Дамерау — Левенштейна
- Алгоритм Нидлмана — Вунша
- Алгоритм Смита — Вотермана
- Soundex
- Metaphone

Деревья для строковых последовательностей

- Суффиксное дерево
- Алгоритм Мак-Крейта
- Алгоритм Укконена
- Алгоритм Вайнера
- Алгоритм Фрача
- Суффиксный массив

Алгоритмы сортировки

- Bogosort
- Stooge sort
- Timsort — гибридный алгоритм сортировки, сочетающий сортировку вставками и сортировку слиянием
- Наивная сортировка — генерация всех $n!$ возможных перестановок и проверка на отсортированность
- Блинная сортировка
- Блочная сортировка (также известен как *корзинная сортировка*), ср. с поразрядной
- Быстрая сортировка — с разбиением исходного набора данных на две половины так, что любой элемент первой половины упорядочен относительно любого элемента второй половины; затем алгоритм применяется рекурсивно к каждой половине
- Гномья сортировка — имеет общее с сортировкой пузырьком и сортировкой вставками. Сложность алгоритма — $O(n^2)$.
- Пирамидальная сортировка (Сортировка кучей) — превращаем список в кучу, берём наибольший элемент и добавляем его в конец списка
- Плавная сортировка
- Поразрядная сортировка — сортирует строки буква за буквой.
- Сортировка Бентли — Седжвика (англ. *BeSe sort*) — модификация быстрой сортировки для составных ключей, заключающаяся в делении не пополам, а на три части — в третью попадают одинаковые (по текущему символу) ключи
- Сортировка с помощью двоичного дерева (англ. *Tree sort*)
- Сортировка методом вставок — определяем, где текущий элемент должен находиться в отсортированном списке, и вставляем его туда
- Сортировка методом выбора — наименьшего или наибольшего элемента и помещения его в начало или конец отсортированного списка
- Сортировка перемешиванием (Сортировка коктейлем)
- Сортировка подсчётом — используется диапазон входных данных, подсчитывается число одинаковых элементов (3 варианта)

- [Сортировка пузырьком](#)
- [Сортировка расчёской](#)
- [Сортировка слиянием](#) — сортируем первую и вторую половину списка отдельно, а затем — сливаем отсортированные списки
- [Сортировка Шелла](#) — попытка улучшить сортировку вставками
- [Топологическая сортировка](#)
- [Хитрая сортировка](#) — извлекает из исходной последовательности отсортированные подпоследовательности, производя их слияние с уже извлечёнными данными
- [Цифровая сортировка](#) — то же, что и [Поразрядная сортировка](#).

Алгоритмы слияния

- [Простой алгоритм слияния](#) (англ. *Simple Merge algorithm*)
- [k-мерный алгоритм слияния](#) (англ. *k-way Merge algorithm*)

Минимизация булевых функций

- [Алгоритм Квайна](#)
- [Алгоритм Мак-Класки](#)
- [Карты Карно](#)
- [Алгоритм Свободы](#)
- [Алгоритм Хва](#)
- [Получение простых импликант на основе разбиения по кодам разности](#)
- [Метод поразрядного выращивания](#)
- [Метод Блейка](#)

Алгоритмы сжатия данных

Алгоритмы сжатия без потерь

- [Преобразование Барроуза — Уилера](#) (также известен как англ. *BWT*) — предварительная обработка данных для улучшения сжатия без потерь
- [Преобразование Шиндлера](#) (англ. *ST*) — модификация преобразования Барроуза — Уилера
- [Алгоритм DEFLATE](#) — популярный свободный алгоритм сжатия (используется в библиотеке zlib)
- [Дельта-кодирование](#) — эффективно для сжатия данных с часто повторяющимися последовательностями
- [Инкрементное кодирование](#) — дельта-кодирование, применяемое к последовательности строк
- Семейство алгоритмов словарного сжатия Лемпеля — Зива:
 - [LZ77](#) — родоначальник семейства LZ77-алгоритмов
 - [LZ77-PM](#)
 - [LZFG](#)
 - [LZFG-PM](#)
 - [LZP](#)

- [LZBW](#)
- [LZSS](#)
 - [LZB](#)
 - [LZH](#)
 - [LZRW1](#)
- [LZ78](#) — родоначальник семейства LZ78-алгоритмов
 - [Алгоритм Лемпеля — Зива — Велча](#) (также известен как [англ. LZW](#)) — сжатие без потерь
 - [LZW-PM](#)
 - [LZMW](#)
 - [LZMA](#) — сокращение от [англ. Lempel-Ziv-Markov chain-Algorithm](#)
 - [LZO](#) — алгоритм сжатия, ориентированный на скорость
- [Алгоритм сжатия PPM](#)
- [Кодирование длин серий](#) (Групповое кодирование, также известен как [англ. RLE](#)) — последовательная серия одинаковых элементов заменяется на два символа: элемент и число его повторений
- [Алгоритм SEQUITUR](#) — сжатие без потерь, автоматическое адаптивное построение контекстно-свободной грамматики для обрабатываемых данных
- [Вейвлет-кодирование на основе вложенных нуль-деревьев \(EZW-кодирование\)](#)
- [Энтропийное кодирование](#) — схема кодирования, которая присваивает коды символам таким образом, чтобы соотнести длину кодов с вероятностью появления символов
 - [Алгоритм Шеннона — Фано](#) — самый простой алгоритм кодирования
 - [Алгоритм Хаффмана](#) — алгоритм построения кода при помощи кодовых деревьев
 - [Адаптивное кодирование Хаффмана](#) — техника адаптивного кодирования, основывающаяся на коде Хаффмана
 - [Усечённое двоичное кодирование](#) — используется для однородного вероятностного распределения с конечным алфавитом
 - [Арифметическое кодирование](#) — развитие энтропийного кодирования
 - [Адаптивное арифметическое кодирование](#) — техника адаптивного кодирования, основывающаяся на арифметическом кодировании
 - [Кодирование расстояний](#) — метод сжатия данных, который близок по эффективности к арифметическому кодированию
- [Энтропийное кодирование с известными характеристиками](#)
 - [Унарное кодирование](#) — код, который представляет число n в виде n единиц с замыкающим нулём
 - [дельта|гамма|омега-кодирование Элиаса](#) ([англ. Elias coding](#)) — универсальный код, кодирующий положительные целые числа
 - [Кодирование Фибоначчи](#) — универсальный код, который кодирует положительные целые числа в двоичные кодовые слова
 - [Кодирование Голомба](#) — форма энтропийного кодирования, которая оптимальна для алфавитов с геометрическим распределением
 - [Кодирование Райса](#) — форма энтропийного кодирования, которая оптимальна для алфавитов с геометрическим распределением

Алгоритмы сжатия с потерями

- Линейное предсказывающее кодирование — сжатие с потерями, представляющее спектральную огибающую цифрового сигнала речи в сжатом виде
- A-закон — стандартный алгоритм компандирования. Применяется в РФ.
- Мю-закон — стандартный алгоритм компандирования
- Фрактальное сжатие — метод, использующий фракталы для сжатия изображений
- Трансформирующее кодирование — тип сжатия данных для «естественных» данных, таких как аудиосигналы или фотографические изображения
- Векторное квантование — техника, часто используемая в сжатии данных с потерями
- Вейвлетное сжатие — тип компрессии данных, хорошо подходящий для сжатия изображений (иногда также используется для сжатия видео и аудио)

Вычислительная геометрия

- Алгоритм Гилберта — Джонсона — Кёрти — определение наименьшего расстояния между двумя выпуклыми множествами
- Поиск пары ближайших точек — трудоёмкость $O(n \log n)$.
- Поиск диаметра множества точек
- Алгоритм Кируса — Бека — отсечение отрезков выпуклым многоугольником.
- Алгоритм Сазерленда — Ходжмана — отсечение многоугольника.
- Построения контура прямоугольников (стороны параллельны осям координат).
- Нахождение ядра многоугольника
- Регуляризация многоугольника — декомпозиция многоугольника на монотонные части.

Построение выпуклой оболочки набора точек

- Построение ВП через треугольники — трудоёмкость $O(n^4)$.
- Построение ВП перебором рёбер на принадлежность — трудоёмкость $O(n^3)$.
- Алгоритм сканирования Грэхема — трудоёмкость $O(n \log n)$.
- Алгоритм Экла — Туссена — трудоёмкость $O(n \log n)$. Улучшение алгоритма Грэхема.
- Алгоритм Эндрю — трудоёмкость $O(n \log n)$. Улучшение алгоритма Грэхема.
- Алгоритм быстрой оболочки — трудоёмкость $O(n^2)$, в среднем — $O(n \log n)$.
- Алгоритм Киркпатрика — построение выпуклой оболочки набора точек на плоскости методом «разделяй и властвуй» через мосты. Трудоёмкость $O(n \log n)$.
- Построение методом «разделяй и властвуй» через построение касательных — трудоёмкость $O(n \log n)$.
- Алгоритм заворачивания подарков (Джарвиса) — трудоёмкость $O(nh)$, h — количество точек в выпуклой оболочке.
- Алгоритм Киркпатрика — Зейделя — трудоёмкость $O(n \log h)$, h — количество точек в выпуклой оболочке.
- Алгоритм Чана — трудоёмкость $O(n \log h)$, h — количество точек в выпуклой оболочке.
- Инкрементальный алгоритм (fast online hull) — через построение касательных $O(n^2)$, с помощью сбалансированного дерева — $O(n \log n)$.
- Приближённая выпуклая оболочка снизу (lower approximate hull) — методом полос. Трудоёмкость $O(nk)$, где k — количество полос.
- Приближённая выпуклая оболочка сверху (upper approximate hull) — методом полос.

Трудоёмкость $O(nk)$, где k — количество полос.

- Алгоритм Ли (выпуклые оболочки) — построение выпуклой оболочки простого многоугольника через отрезание карманов. Трудоёмкость $O(n)$.

Триангуляция

- Триангуляция через поиск диагоналей — ищется диагональ, многоугольник делится на два и далее рекурсивно. Трудоёмкость $O(n^4)$.
- Триангуляция через отрезание ушей — ищется образующая треугольник диагональ, соседние с треугольником вершины — следующие претенденты на отрезание. Трудоёмкость $O(n^2)$.
- Триангуляция монотонного простого многоугольника — трудоёмкость $O(n)$.
- Жадная триангуляция — трудоёмкость $O(n^2 \log n)$.
- Оптимальная триангуляция — NP -полная задача. Суммарная длина всех рёбер минимальна среди всех триангуляций данного множества.

Триангуляция Делоне

- Итеративные алгоритмы построения триангуляции Делоне — трудоёмкость $O(n^2)$.
- Алгоритмы построения триангуляции Делоне слиянием — трудоёмкость $O(n^2)$ и $O(n \log n)$.
- Алгоритмы прямого построения триангуляции Делоне — трудоёмкость $O(n^2)$.
- Двухпроходные алгоритмы построения триангуляции Делоне — трудоёмкость $O(n^2)$ и $O(n \log n)$.
- Триангуляция Делоне с ограничениями — трудоёмкость $O(n^2)$.

Квазитриангуляция

- Алгоритм построения квазитриангуляции

Диаграмма Вороного

- Простой алгоритм построения диаграммы Вороного — трудоёмкость $O(n^2 \log n)$.
- Алгоритм построения диаграммы Вороного через заметающую прямую — трудоёмкость $O(n \log n)$.
- Рекурсивный алгоритм построения диаграммы Вороного — трудоёмкость $O(n \log n)$.

Локализация точки

- Локализация точки для выпуклого многоугольника — время запроса $O(\log n)$.
- Локализация точки в звёздном многоугольнике — время запроса $O(\log n)$.
- Алгоритм точки в многоугольнике — проверка принадлежности данной точки простому многоугольнику $O(n)$.
- Метод луча — принадлежность точки простому многоугольнику $O(n)$.
- Метод углов — принадлежность точки выпуклому многоугольнику. Трудоёмкость

$O(n)$.

- Метод полос — простой многоугольник. Время запроса $O(\log n)$, память $O(n^2)$.
- Метод детализации триангуляции Киркпатрика — простой многоугольник. Время запроса $O(\log n)$, память $O(n)$.
- Трапециoidalная карта — простой многоугольник. Рандомизированный алгоритм, время запроса $O(\log n)$, память $O(n)$.
- Метод цепей — простой многоугольник. Время запроса $O(\log^2 n)$, память $O(n)$.

Пересечения

- Алгоритм Бентли — Оттманна — поиск всех точек пересечения отрезков на плоскости $O((n + k) \log n)$, k — количество точек пересечения.
- Алгоритм Чазелла — Эдельсбруннера — пересечение отрезков за $O(k + n \log n)$.
- Определение наличия пересекающихся отрезков (алгоритм Шеймоса — Гоя) — трудоёмкость $O(n \log n)$.
- Алгоритм Козна — Сазерленда — для выпуклых многоугольников. Трудоёмкость $O(n_1 n_2)$.
- Пересечения выпуклых многоугольников — трудоёмкость $O(n_1 + n_2)$.
- Алгоритм Шеймоса — Хоуи — для выпуклых многоугольников методом полос. Трудоёмкость $O(n_1 + n_2)$.
- Пересечения выпуклых многоугольников с заметающей прямой — трудоёмкость $O(n_1 + n_2)$.
- Пересечение звёздных многоугольников — трудоёмкость $O(n_1 n_2)$.
- Пересечение полуплоскостей — трудоёмкость $O(n \log n)$.
- Алгоритм Лианга — Барски
- Быстрое отсечение
- Алгоритм Кируса — Бека
- Николла — Ли — Николла
- Алгоритм Сазерленда — Ходжмана
- Алгоритм Уайлера — Атертона

Вращающиеся калиперы

- Поиск диаметра множества точек через вращающиеся калиперы
- Поиск минимального по площади описанного прямоугольника для множества точек
- Поиск минимального по периметру описанного прямоугольника для множества точек
- Определение ширины многоугольника
- Построение суммы Минковского двух выпуклых многоугольников
- Поиск максимального расстояния между двумя множествами точек
- Поиск минимального расстояния между двумя выпуклыми многоугольниками
- Построение мостов для двух выпуклых многоугольников
- Построение критических опорных прямых для выпуклых многоугольников

Компьютерная графика

- Алгоритмы построения отрезка — алгоритмы для аппроксимации отрезка на

дискретной графической поверхности

- Алгоритм Брезенхэма — растеризует отрезок линии с заданными координатами начала и конца
- Алгоритм DDA-линии — чертит точки двумерного массива в форме прямой линии между двумя заданными точками (использует вычисления с плавающей точкой)
- Алгоритм Ву — алгоритм растеризации отрезка со сглаживанием
- Алгоритм закраски с затравкой — заполняет соединённый регион многомерного массива указанным значением
- Алгоритм художника — определяет видимые части трёхмерной сцены
- Трассировка лучей — рендеринг реалистичных изображений
- Модель освещения
 - Затенение по Фонгу — модель освещения и метод интерполяции в трёхмерной компьютерной графике
 - Затенение по Гуро — алгоритм моделирования различных эффектов света и цвета на поверхности объекта в трёхмерной компьютерной графике
 - Модель освещения Блинна — Фонга
 - Модель освещения Кука — Торренса
- Алгоритм сканирующей строки (англ. *Scanline rendering*) — конструирует образ с помощью перемещения воображаемой линии над образом
- Глобальное освещение — рассматривает прямое освещение и отражение от других объектов
- Алгоритмы интерполяции — конструирование новых точек данных, таких как в цифровом увеличителе
 - Интерполяция сплайнами — уменьшение ошибки феномена Рунге

Компьютерное зрение

- Epitome — представление образа или видео при помощи меньшего образа или видео

Криптографические алгоритмы

*См. также Разделы в криптографии для **аналитического глоссария***

- Шифрование с симметричным (скрытым) ключом:
 - ГОСТ 28147-89
 - AES (англ. *Advanced Encryption Standard*) — победитель соревнования NIST, также известен как Rijndael
 - Blowfish
 - DES (англ. *Data Encryption Standard*) — иногда, алгоритм DEA (англ. *Data Encryption Algorithm*), победитель соревнования NBS, заменён на AES для большинства применений
 - RC2
 - IDEA (англ. *International Data Encryption Algorithm*)
 - RC4

- Асимметричное шифрование (с публичным ключом)
 - Алгоритм Эль-Гамала
 - RSA
 - NTRUEncrypt
- Алгоритмы выработки общего ключа
 - Алгоритм Диффи — Хеллмана
- Алгоритмы цифровой подписи:
 - ГОСТ Р 34.10-94 — устаревший российский стандарт цифровой подписи, модификация схемы Эль-Гамала
 - ГОСТ Р 34.10-2001 — российский стандарт цифровой подписи, основанный на эллиптических кривых
 - DSA (англ. *Digital Signature Algorithm*) — базируется на схеме Эль-Гамала
 - ECDSA (англ. *Elliptic Curve Digital Signature Algorithm*) — перенос DSA на эллиптические кривые
- Алгоритмы разделения секрета
 - Рюкзак — на данный момент доказана нестойкость схемы
 - Схема Шамира
 - Схема Blakey
- Алгоритмы подбрасывания монеты по телефону
- Доказательство с нулевым разглашением
- Криптографические функции дайджестов сообщений:
 - ГОСТ Р 34.11-94
 - MD5 Резюме сообщения 5 (Message Digest 5) Разработан Рональдом Ривестом (RFC 1321) — существует метод генерации коллизий
 - RIPEMD-160
 - SHA-1
 - HMAC — аутентификация сообщение с помощью хеш-ключа
 - Тигр — обычно используется в ТТН
- Криптостойкие генераторы псевдослучайных чисел
 - Алгоритм Блюма — Блюма — Шуба — базируется на сложности факторизации
 - Алгоритм Ярроу
 - Алгоритм Fortuna — улучшение алгоритма Ярроу
 - Генерация случайных простых чисел
 - Алгоритм аутентификации сообщений Message authentication algorithm

Цифровая обработка сигналов

- CORDIC — быстрая техника вычисления тригонометрических функций.
- Медианный фильтр для одномерного массива
- Дождевой алгоритм — уменьшает комплексную историю давлений в расчёте элементарных противодействий для использования в анализе усталости

- Osem — алгоритм для обработки медицинских изображений
- Алгоритм Гёрцеля — может быть использован для декодирования цифр тональных сигналов
- Развеевание Ричардсона — Люси — алгоритм увеличения резкости образа

Разработка программного обеспечения

- Алгоритмы для восстановления и изоляции повреждённых семантик
- Алгоритм сравнения Unicode
- Алгоритм преобразования CHS — Преобразование между системами адресации диска
- Алгоритм вычисления контрольной суммы (CRC или FCS) Циклическая избыточная сумма (англ. *Cyclic Redunancy Check*), или контрольная последовательность кадра (англ. *Frame Check Sequence*) — вычисление кода проверки.
- Чётность — Проверка четности количества единиц в двоичной записи числа. Позволяет обнаруживать ошибку в одном разряде.
- Алгоритм соединения (СУБД) — реализация операции соединения реляционной алгебры.

Алгоритмы распределённых систем

- Упорядочение Лампорта — Частичное упорядочение событий в зависимости от *того, что случилось раньше*
- Алгоритм мгновенного снимка — снимок процесса, записывающий глобальное состояние системы
- Векторное упорядочение — Полное упорядочение событий
- Алгоритм Марзулло — распределённая синхронизация часов
- Алгоритм пересечений — другой алгоритм синхронизации часов

Алгоритмы выделения и освобождения памяти

- Сборщик мусора Боема — «скромный» сборщик мусора
- Дружеское выделение памяти — алгоритм выделения памяти таким образом, чтобы фрагментация была наименьшей.
- Сборщик мусора с поколениями — быстрые сборщики мусора, которые разделяют память по возрасту
- Пометить и выместить
- Подсчёт ссылок

Алгоритмы в операционных системах

- Алгоритм банкира — Алгоритм, использующийся для избежания взаимных блокировок
- Алгоритм замены страницы — выбор страницы-жертвы при условиях небольшого объёма памяти
- Адаптивный алгоритм замещения кэша: скорость выполнения лучше, чем у LRU
- Часы с адаптивной заменой (CAR): алгоритм замены страниц со скоростью выполнения, сравнимой с адаптивным алгоритмом замещения кэша
- Алгоритм забияки — выбор нового лидера среди множества компьютеров

- rsync — алгоритм, использующийся для эффективной передачи файлов между двумя компьютерами

Дисковые алгоритмы-планировщики

- Алгоритм лифта — дисковый алгоритм планирования, который работает как лифт
- Алгоритм кратчайшего перемещения — дисковый алгоритм планирования для уменьшения времени поиска

Сетевые алгоритмы

- Алгоритм Карна: получение точных оценок времени распространения пакетов сообщений при использовании TCP/IP
- Алгоритм Лулео: техника эффективного сохранения и поиска в таблицах роутинга
- Нагрузка на сеть
 - Экспоненциальная задержка
 - Алгоритм Нагла: улучшение эффективности TCP/IP за счёт объединения пакетов
 - Усечённая бинарная экспоненциальная задержка
- Шейпинг
 - Алгоритм текущего ведра
 - Алгоритм ведра маркеров

Алгоритмы синхронизации процессов

- Алгоритм Петерсона
- Алгоритм пекарни Лампорта
- Алгоритм Деккера

Алгоритмы планирования

- Планирование с постоянной скоростью
- Первый заканчивает раньше (планирование)
- Честное разделение (планирование)
- Планирование Round-robin
- Многоуровневая отдача (планирование) (англ. *Multi level feedback queue*)
- Кратчайшее оставшееся время (планирование)
- Наименьшее время бездействия (планирование)
- Списковое планирование (англ. *List scheduling*)
- Алгоритм высоких вершин
- Кратчайшая работа следующей
- Кратчайшее оставшееся время

Генетические алгоритмы

- Выбор пропорционально пригодности — также известен как **выбор рулеточного**

колеса

Медицинские алгоритмы

- [Медицинский алгоритм](#)
- [Техасский проект медицинских алгоритмов](#)

Нейронные сети

- [Метод обратного распространения ошибки](#)
- [Самоорганизующееся отображение](#) (Карты Кохонена, SOM)
- [Метод коррекции ошибки](#)
- [Метод коррекции с обратной передачей сигнала ошибки](#)

Вычислительная алгебра

- [Алгоритм Бухбергера](#) — находит [базис Грёбнера](#)
- [Процесс Грама — Шмидта](#) — ортогонализация набора векторов
- [Алгоритм пополнения Кнута — Бендикса](#)
- [Алгоритм мультивариационного деления](#) — для [многочленов](#) в некоторых неопределённостях
- [Алгоритмы умножения матриц](#)
 - [Алгоритм Штрассена](#) — быстрое умножение матриц
 - [Алгоритм Копперсмита](#) — Винограда
- [Умножение цепных матриц](#) (англ. *Chain matrix multiplication*)
- [Алгоритм Катхилла — Макки](#) — алгоритм уменьшения ширины ленты разреженных симметричных матриц
- [Алгоритмы вычисления дискретного преобразования Фурье](#)
 - [Быстрое преобразование Фурье](#)
 - [Алгоритм БПФ Кули — Туки](#)
 - [Алгоритм БПФ Рэдера](#)
 - [Алгоритм БПФ Блюштейна](#)
 - [Алгоритм БПФ Брууна](#)
 - [Алгоритм БПФ при помощи простых сомножителей](#)
- [Алгоритм нахождения собственного значения матрицы](#)
- [Преобразования Хаусхолдера \(*QR*-разложение\)](#) — вычисление обратной матрицы, собственных векторов и собственных значений матрицы; используется также для решения систем линейных уравнений.
- [Решение систем линейных уравнений](#)
 - [Метод Гаусса](#) (Гауссово исключение) — стандартный метод решения систем линейных уравнений
 - [Структурированное гауссово исключение](#) — применяется, когда матрица системы является разреженной
 - [Метод Жордана — Гаусса](#) — модификация метода Гаусса для матричного представления
 - [Разложение Холецкого](#) — метод, эффективный для ленточных и разреженных матриц

- Метод Пранис — Праневича — решение систем линейных уравнений с параллельными вычислениями по компонентам

Теоретико-числовые алгоритмы

- Целочисленная арифметика (алгоритмы для работы с большими числами)
 - Умножение столбиком больших чисел
 - «Быстрый столбик»
 - Умножение Карацубы — алгоритм быстрого умножения чисел
 - Алгоритм Тоома — Кука — обобщённый алгоритм умножения Карацубы (известен также как Toom-3)
 - Метод умножения Шёнхаге — Штрассена — более быстрый алгоритм умножения
 - Алгоритм Фюрера — на данный момент самый быстрый алгоритм умножения больших чисел
 - Деление на одноразрядное число (DO)
 - Деление больших чисел
- Быстрое возведение в степень — вычисляет степени чисел при помощи возведения в квадрат
- Алгоритмы модулярной арифметики
 - Алгоритм Монтгомери — модулярное умножение и возведение в степень
 - Алгоритм нахождения порядка элемента
 - Алгоритм Тонелли — Шенкса — решение квадратичных сравнений
 - Решение систем линейных сравнений
 - С помощью китайской теоремы об остатках
 - Алгоритм Гарнера
- Решение систем линейных уравнений над полем
 - Алгоритм Ланцоша — эффективен над полем характеристики 2
 - Алгоритм Видемана
- Дискретное логарифмирование:
 - В простом конечном поле
 - Алгоритм Шенкса (алгоритм больших и маленьких шагов, англ. *baby-step giant-step*)
 - Алгоритм Полига — Хеллмана — эффективен, если все делители $p - 1$ — небольшие
 - p -метод Полларда дискретного логарифмирования
 - Алгоритм Адлемана — первый субэкспоненциальный алгоритм дискретного логарифмирования
 - Алгоритм COS (алгоритм Копперсмита — Одлышко — Шреппеля) — достаточно эффективный субэкспоненциальный алгоритм
 - Решето числового поля — наиболее эффективный на данный момент алгоритм дискретного логарифмирования
 - В произвольном конечном поле
 - Алгоритм исчисления индексов (алгоритм index-calculus) — сведение дискретного логарифмирования в произвольном конечном поле к аналогичной задаче в простом поле
 - Алгоритм Копперсмита — эффективный алгоритм дискретного

логарифмирования в конечном поле характеристики 2

- Алгоритмы нахождения наибольшего общего делителя (НОД) двух чисел
 - Алгоритм Евклида
 - Расширенный алгоритм Евклида — также решает уравнение $ax + by = c$, где $c = \text{НОД}(a, b)$, $\text{НОД}(a, y) = \text{НОД}(b, x) = 1$
 - Бинарный алгоритм вычисления НОД — эффективный способ вычисления НОД
 - Расширенный бинарный алгоритм — модификация бинарного алгоритма нахождения НОД, аналогичная расширенному алгоритму Евклида
- Простые числа:
 - Нахождение простых чисел:
 - Перебор делителей
 - Решето Эратосфена
 - Решето Аткина — оптимизированная версия решета Эратосфена
 - Решето Сундарама
 - Тесты простоты — проверка, является ли данное число простым:
 - Детерминированные тесты простоты:
 - Тест на основе малой теоремы Ферма
 - Тест Миллера — модификация теста на основе малой теоремы Ферма; опирается на расширенную гипотезу Римана
 - (N-1)-метод проверки простоты — тест на простоту при известном разложении на множители числа $N - 1$; также используется для построения больших простых чисел
 - (N+1)-метод проверки простоты — тест на простоту при известном разложении на множители числа $N + 1$
 - Алгоритм Конягина — Померанса — модификация $(N - 1)$ -метода
 - Алгоритм Ленстры — использует суммы Якоби и некоторые тесты, обобщающие малую теорему Ферма
 - Тест Люка — Лемера для чисел Мерсенна
 - Тест Пепина для чисел Ферма
 - Тест Агравала — Каяла — Саксены — полиномиальный детерминированный тест простоты
 - Вероятностные тесты простоты:
 - Тест Соловея — Штрассена — опирается на малую теорему Ферма и свойства символа Лежандра
 - Тест Миллера — Рабина — эффективная модификация теста Соловея — Штрассена
 - Факторизация — разложение числа на простые множители:
 - Алгоритмы с экспоненциальной сложностью:
 - Перебор делителей
 - Метод факторизации Ферма
 - $(p - 1)$ -алгоритм Полларда
 - p -метод Полларда
 - Метод Лемана
 - Алгоритм Ленстры
 - Алгоритмы с субэкспоненциальной сложностью:

- [Алгоритм Диксона](#)
- [Метод квадратичного решета](#) (англ. *QS*)
- [Специальный метод решета числового поля](#) (англ. *SNFS*)
- [Общий метод решета числового поля](#) (англ. *GNFS*)
- [Факторизация с помощью эллиптических кривых](#) — вероятностный алгоритм факторизации с помощью [эллиптических кривых](#)
- [Дроби](#)
 - [Жадный алгоритм для египетских дробей](#) — преобразует рациональные числа в египетские дроби
- [Прочие](#)
 - [Алгоритм Шуфа](#) — вычисление порядка группы точек эллиптической кривой
 - [Алгоритм Ленстры](#) — [Ленстры](#) — [Ловаса](#) (LLL-алгоритм, L^3 -алгоритм)

Численные алгоритмы

- [Алгоритм де Кастельжо](#) — вычисление [кривых Безье](#)
- [Методы интерполяции](#)
 - [Линейное сглаживание по трём точкам](#)
 - [Линейное сглаживание по пяти точкам](#)
 - [Нелинейное сглаживание по семи точкам](#)
- [Приближенное вычисление решений](#)
 - [Метод фальшпозиции](#) (*False position method, regula falsi method*) — аппроксимирует корни функции
 - [Метод бисекции](#) — нахождение нуля (корня) нелинейной функции
 - [Метод Ньютона](#) (метод касательных) — нахождение нулей функций с помощью производной
 - [Метод секущих](#) (метод хорд) — аппроксимирует корни функции
 - [Метод градиентов](#) ([градиентный спуск](#)) — аппроксимирует решение системы уравнений
 - [Метод сопряжённого градиента](#)
 - [Алгоритм Гаусса — Ньютона](#) — алгоритм для решения нелинейных уравнений методом наименьших квадратов
 - [Алгоритм Левенберга — Марквардта](#) — алгоритм для решения нелинейных уравнений [методом наименьших квадратов](#)
- [Танцующие звенья](#) — нахождение всех решений задачи точного покрытия
- [Алгоритм де Бора](#) — вычисление [сплайнов](#)
- [Алгоритм Гаусса — Лежандра](#) — вычисление цифр числа [пи](#)
- [Алгоритм Кэхэна](#) — более аккуратный метод суммирования чисел с плавающей точкой
- [Алгоритм MISER](#) — моделирование [методом Монте-Карло](#), [численное интегрирование](#)
- [Функции округления](#) — классические способы округления чисел
- [Сдвигающий алгоритм корня n-ой степени](#)[Шаблон:Ненп](#) — извлечение корня цифра за цифрой
- [Вычисление квадратного корня](#):
 - [Алгоритм Герона](#)

- школьный (ручной) алгоритм — аппроксимирует квадратный корень числа
- Вычисление корня n -ной степени

Алгоритмы оптимизации

- Линейное программирование
- Симплекс-метод
- «Венгерский метод» — решение задач целочисленного линейного программирования
- Метод Мака решения задачи о назначениях
- Алгоритм имитации отжига
- Метод роя частиц
- Муравьиные алгоритмы
- Метод ветвей и границ
- Дифференциальная эволюция
- Эволюционная стратегия
- Метод Нелдера — Мида (downhill simplex method) — алгоритм нелинейной оптимизации
- Всхождение со случайным перезапуском
- Стохастическое туннелирование
- Задача о сумме подмножеств
- Метод перебора
- Метод Фибоначчи поиска экстремума — метод выбора точек для нахождения экстремума функции одной переменной
- Градиентный спуск
- Алгоритм Левенберга — Маркардта — комбинация метода Ньютона и наискорейшего спуска
- Метод Ньютона в оптимизации, основанный на методе Ньютона поиска корня
- Квазиньютоновские методы — методы, основанные на замене матрицы Гессе её приближением.
 - Алгоритм Бroyдена — Флетчера — Гольдфарба — Шанно — квазиньютоновский алгоритм нелинейной оптимизации

Грамматический разбор

- Рекурсивный нисходящий парсер — нисходящий парсер, подходящий для $LL(k)$ -грамматик
- LL-парсер — относительно простой алгоритм разбора за линейное время для ограниченного класса контекстно-свободных грамматик
- LR-парсер — более сложный алгоритм разбора за линейное время для большего класса контекстно-свободных грамматик. Варианты:
 - Парсер первенства операторов
 - SLR-парсер (англ. *Simple LR parser*)
 - LALR-парсер (англ. *Look-ahead LR parser*)
 - Канонический LR-парсер
- Парсер старьёвщика — алгоритм разбора за линейное время, поддерживающий некоторые контекстно-свободные грамматики и грамматики разбора выражений
- Алгоритм Коука — Янгера — Касами (алгоритм CYK) — $O(n^3)$ -алгоритм для разбора

любой контекстно-свободной грамматики

- Алгоритм Эрли — другой $O(n^3)$ -алгоритм для разбора любой контекстно-свободной грамматики
- GLR-парсер — алгоритм для разбора любой контекстно-свободной грамматики, он предназначен для определённых грамматик, на которых он действует практически за линейное время и $O(n^3)$ в худшем случае.
- Метод рекурсивного спуска — это один из методов определения принадлежности входной строки к некоторому формальному языку, описанному контекстно-свободной грамматикой
- Алгоритм Рутисхаузера — работает в предположении о полной скобочной структуре выражения
- Алгоритм сортировочной станции

Квантовые алгоритмы

Приложения квантовых вычислений к различным категориям проблем и алгоритмы

- Алгоритм Гровера — позволяет выполнять поиск среди N вариантов за $O(\sqrt{N})$ проверок
- Алгоритм Шора — полиномиальный алгоритм факторизации числа
- Алгоритм Дойча — Йожи — критерий баланса для булевой функции
- Задача Фейнмана

Теория вычислений и автоматов

- Конструирование набора подмножеств — алгоритм для преобразования недетерминированного автомата в детерминированный
- Алгоритм Тодда — Коксетера — процедура для создания сомножеств

Другие

- Астрономические алгоритмы
- Алгоритмы восстановления фазы волнового фронта (Гершберга-Сакстона, Фиенапа, Бейтса, Альморо и др.)
- Алгоритм Баума — Велша
- Алгоритмы манипуляции битами
 - Алгоритм создания битовой маски
 - Алгоритм обмена при помощи исключающего ИЛИ — обмен значений двух переменных без использования буфера
- Алгоритм вычисления дня недели (Алгоритм Судного Дня)
- Алгоритм Шрейера — Симса
- Алгоритм Витерби
- Алгоритм Луна — метод проверки правильности идентификационных чисел
- Алгоритм стемматизации (стемминга) — процесс нахождения основы слова
- Алгоритм Риша — нахождение первообразных
- Алгоритм синтеза многосвязной сети
- Алгоритм распространения доверия (алгоритм «sum-product») — алгоритм маргинализации с помощью двунаправленной передачи сообщений на графе, применяемый для вывода на графических вероятностных моделях

- Быстрый метод мультиполей — алгоритм вычисления сил взаимодействия N частиц^[2]
- Алгоритм Зиккурат — алгоритм генерации неравномерно распределённых случайных чисел

См. также

- Список структур данных
- Список классов сложности, Класс сложности
- Список основных разделов теории алгоритмов

Примечания

1. В тематическом проекте есть также список терминов, относящихся к алгоритмам и структурам данных, составленный на основе словаря Американского национального института стандартов. Если Вы планируете добавить какой-либо алгоритм в этот список, убедитесь, пожалуйста, что его здесь ещё нет (возможно, алгоритм упоминается под каким-либо альтернативным названием). Внимательно посмотрите, к какой именно категории относится данный алгоритм. В случае, когда из названия не ясно, что именно делает алгоритм, напишите, пожалуйста, краткое описание. Если Вы планируете написать статью про один из алгоритмов, упомянутых в этом списке, пожалуйста, прочитайте сначала руководство «Википедия:Алгоритмы в Википедии» или посмотрите несколько уже написанных статей, посвящённых алгоритмам.
2. *Barry A. Cipra*. The Best of the 20th Century: Editors Name Top 10 Algorithms (<http://www.siam.org/news/news.php?id=637>) (англ.) // SIAM News. — 2000. — Vol. 33, no. 4. Архивировано (<https://web.archive.org/web/20110615092015/http://www.siam.org/news/news.php?id=637>) 15 июня 2011 года.

Литература

- *Ахо, Альфред, В., Хопкрофт, Джон, Ульман, Джеффри, Д.* Структуры данных и алгоритмы. — Издательский дом «Вильямс», 2000. — 384 с. — ISBN 5-8459-0122-7 (рус.) / ISBN 0-201-00023-7 (англ.).
- *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии (<http://www.ict.edu.ru/ft/002416/book.pdf>) . — Москва: МЦНМО, 2003. — 328 с. — ISBN 5-94057-103-4.
- *Дональд Кнут*. Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, Volume 1. Fundamental Algorithms. — 3-е изд. — М.: «Вильямс», 2006. — 720 с. — ISBN 5-8459-0080-8.
- *Дональд Кнут*. Искусство программирования, том 1, выпуск 1. MMIX — RISC-компьютеры нового тысячелетия = The Art of Computer Programming, Volume 1, Fascicle 1: MMIX — A RISC Computer for the New Millennium. — М.: «Вильямс», 2007. — 160 с. — ISBN 978-5-8459-1163-6.
- *Дональд Кнут*. Искусство программирования, том 2. Получисленные методы = The Art of Computer Programming, Volume 2. Seminumerical Algorithms. — 3-е изд. — М.: «Вильямс», 2007. — 832 с. — ISBN 5-8459-0081-6.
- *Дональд Кнут*. Искусство программирования, том 3. Сортировка и поиск = The Art of Computer Programming, Volume 3. Sorting and Searching. — 2-е изд. — М.: «Вильямс», 2007. — 824 с. — ISBN 5-8459-0082-4.
- *Дональд Кнут*. Искусство программирования, том 4, А. Комбинаторные алгоритмы, часть 1 = The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1. — М.: «Вильямс», 2013. — 960 с. — ISBN 978-5-8459-1744-7.

- *Д-р Сидни Фейт*. TCP/IP: Архитектура, протоколы, реализация (включая IP версии 6 и IP Security) = TCP/IP: Arhitecture, Protocols, and Implementation with IPv6 and IP Security. — 2nd. ed. Dr. Sidnie Feit Copyright 1997, 1993 by The McGraw-Hill Companies, Inc. (включая IP версии 6 и IP Security). — 2-е изд. — М.: Издательство «Лори», 2003. — 424 с. — ISBN 5-85582-072-6 (рус.) / ISBN 0-07-021389-5 (англ.).
- *Порублев Илья Николаевич, Ставровский Андрей Борисович*. Алгоритмы и программы. Решение олимпиадных задач. — М.: «Вильямс», 2007. — 480 с. — ISBN 978-5-8459-1244-2.
- *Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн*. Алгоритмы: построение и анализ, 3-е издание = Introduction to Algorithms, Third Edition. — М.: «Вильямс», 2013. — 1328 с. — ISBN 978-5-8459-1794-2.
- *Роберт Седжвик*. Фундаментальные алгоритмы на С. Анализ/Структуры данных/Сортировка/Поиск = Algorithms in C. Fundamentals/Data Structures/Sorting /Searching. — СПб.: ДиаСофтЮП, 2003. — 672 с. — ISBN 5-93772-081-4.
- *Роберт Седжвик*. Фундаментальные алгоритмы на С. Алгоритмы на графах = Algorithms in C. Graph Algorithms. — СПб.: ДиаСофтЮП, 2003. — 480 с. — ISBN 5-93772-082-2.
- *Sanjoy Dasgupta, Christos H. Papadimitriou, Umesh Vazirani*. Algorithms (<http://www.cs.berkeley.edu/~vazirani/algorithms.html>). — The McGraw-Hill Companies, 2006. — 320 с. — ISBN 0-07-352340-2.
- *Ричард Берд*. Жемчужины проектирования алгоритмов. Функциональный подход = Pearls of Functional Algorithm Design. — ДМК Пресс, 2013. — (Функциональное программирование). — ISBN 978-5-94074-867-0.
- *Препарата Ф., Шеймос М*. Вычислительная геометрия: Введение = Computational Geometry An introduction. — М.: Мир, 1989. — 478 с.
- *Берг М., Чеонг О., Кревельд М., Овермарс М*. Вычислительная геометрия. Алгоритмы и приложения = Computational Geometry: Algorithms and Applications. — М.: ДМК-Пресс, 2016. — 438 с. — ISBN 978-5-97060-406-9.

Ссылки

- Сайт по методам сжатия данных (<http://www.compression.ru/>)
 - Дискретная математика: Алгоритмы (<http://rain.ifmo.ru/cat/view.php/>) — библиотека алгоритмов и визуализаторов
 - Алгоритмы, методы, исходники (<http://algolist.manual.ru/>)
 - Множество алгоритмов с примерами их реализации (<http://e-maxx.ru/algo/>)
-

Источник — https://ru.wikipedia.org/w/index.php?title=Список_алгоритмов&oldid=100357523

Эта страница в последний раз была отредактирована 11 июня 2019 в 10:59.

Текст доступен по [лицензии Creative Commons Attribution-ShareAlike](#); в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.