Введение в теорию алгоритмов

Несколько примеров интуитивного понятня алгоритма:

- Алгоритм — это понятные и точные предписания исполнителю совершить конечное число шагов, направленных на решение поставленной задачи
- Алгоритм — точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время
- Алгоритм — это конечный набор правил, который определяет последовательность операций для решения конкретного множества задач и обладает пятью важными чертами: конечность определённость, ввод, вывод, эффективность. (Д. Кнут)
- Алгоритм — это точное предписание, определяюее вычислитлельный процесс.

Основные свойства алгоритмов

- Дискретность
- Детерминированность (определённость)
- Понятность
- Завершаемость (результативность, конечность)
- Массовость (универсальность) — пусть задача может не иметь универсального алгортима решения для всего класса задач, есть алгоритмы для подмножества
- Однозначность результата

Оказывается, существуют такие классы задач, для решения которых не может быть единого универсального алгоритма.

Проблемы такого рода называют **алгоритмически неразрешимыми проблемами**.

Неразрешимая ппроблема — это та, для которой никогда не может быть найден алгоритм поиска решения, который всегда будет давать правильнер решение для каждого входного значения.

Примеры неразрешимых задач:

1. **Проблема остановки** — это проблема определения результативности алгоритма по его тексту и входным данным. Нет общего алгориома для отладки программ, который по тексту любой программы и её данным определял бы, зациклится программа на этих данных или нет, т. е. не существует универсального алгоритма, позволяющео определиТь, будет ли алгоритм коректно работать с заданными начальными данными.
2. **Проблема распеделения девяток в числе π**. Функция $h(n)$, которая $\forall n \ = 1$. Еслли в десятичной записи числа π есть n стоящих подряд девяток, окржённых другими цифрами, и равна нулю, если такой цепочки девяток нет. Поскольку число π является иррациональным и трансцендентным, то нет никакой информации о распределении девяток (равно как и любых лдругих цифр) в десятичной записи числа. Вычислени функции $h(n)$ идё до тех пор, пока не обнаружится n девяток подряд, однако никакого общего методда вычисления нет, поэтому для некоторых n вычисления могут продолжаться бесконечно.
3. **Нахождение совершенных чисел**, которые равны сумме своих делителей. Дана функция $S(n)$, которая по произвольно заданному числу n ищет n-е по счёту совершенное число. Не существует общего метода вычисления такой функции. Неизвестно даже, счётно или конечно множество совершенных чисел, поэму алгоритм должен перебирать все числа подряд, проверяя их на совершенность. И неизвестно, существует ли следующее число.

4. **Десятая проблема Гильберта** (1900 г.). Нет алгоритма решения произвольных лгебраических диофантовых уравнений вида $P(X_1, X_2, ..., X_n) = 0$, где P — целочисленная функция (например, полином с нецелыми коэффициентами), а переменные X принимают целые значения...
5. **Теорема Гёделя о неполноте формальной арифметики**. Существуют некоторые утверждения, которые не могут быть ни доказаны, ни опровергнуты на основе любого набора непротиворечивых аксиом. ЧТобы доказать эти утверждения, приходится вводить нровые аксиомы. Но как бы не расшимрялас формальная система, осегда найдутся утвержкдения об объектах системы, истинность которых не может быь установлена в рамках самой системы. Поскольку человек может определить, как расширить формальную систему так, чтбы полувчить желаемый результат, то отсюда делается вывод о невозможности формалиации мышления, т. е. мышление невозможно реализовать на компьютере.
6. **Проблема единичной матрицы**. Для данного конечного множества квадратных матриц $n \times n$ определить, существует ли произведение всех или некоторых из этих матриц (возможно. с повторениями) в каком-либо порядке, дающее матрицу. Для разных n существует разная информация о разрешимости.

Таким обраом, в разных радлелаъ математики встречаюьттся **алгоритмически неразрешимые заачи**, т. е. задачи, для которых нет алгоритма решенийя, причём нет не потому что его пока не придумали, а птотму что он невозможетн в принципе.

Основные задачи теории алгоритмов

- Формализация понятия алгоритма, исследование формальных алгоритмическиъх систем
- Формальное доказателльство алгоритмической нерарешимости ряда задач
- Исследование эффецтивности алгортимов

Схема определения понятия алгоритма

Данные формируются из конечного множества символов — алфвита. Из символов алвфавита формируются выражения, с которыми работает алгоритм. Такие последовательности называются словами. В некоторых алгоритмических моделях разрешены любые слова, но не везде — где-то это регулируется грамматикой языка.

Память

Элементарный шаг

Детерменированность

Результативность алгоритм при всех допустимых данных завершает работу

Принято все алгритмические задачи делить на два больших класса:
- Задачи вычисления значения функции
- Задачи распознавания принадлежжности объекта заданному множеству

Первый классзадач в результате выполнния алгоритма предусматривает вычислние значения функции (алфавитное преобразование). В этом случае под задачей понимается $F : \{A\} * \to \{A *\}$, отображаюзая слова из некоторого алфавита в слова этого же алфавита и с этой ункцией связана следюущая алгоритмическая проблема: имеется $X \in A$, нужно найти $F(X)$ Оказалось, чо не всегда можно найти алгор...

ТЕМА ДЛЯ СЕМИНАРА: МОЖНО ЛИ СВЕСТИ ВТОРОЙ ТИП ЗАДАЧ К ПЕРВОЙ?

Основные типы алгоритмиеских моделей

- Алгоритм как некое детерминированное утройство — абстрактные машины. Машина Тьюринга и машина Поста.
- Алгоритм как процедура вычисления некой числовой функции. Рекурсивные функции Черча.
- Алгоритм как последовательность преобразований цепочек в каком-либо алфавите. (Комбинаторные операции над словами). Нормальные алгоритмы Маркова.

Машина Поста

- Тезис Поста — всякий алгоритм представим в форме машины Поста
- Алгоритм (по Посту) — программа для машины Поста, приводящая к решению поставленной задачи

Пост умник конечно: алгоритм — это программа для машины Поста => всякий алгоритм представим в форме машины Поста, чтд

Пример: покажем, как можно воспользоваться командой условногор перехода для организации циклического процесса. Пусть на ленте имеетсящапись из нескольких метод подряд, и головка находится над самой крайней меткой справа. Требуется перевести головку влево до первой пустой позиции.

$$1 \leftarrow 2$$
$$2?\ 3; 1$$
$$3!$$