Элементы теории вычислительной сложности

2 вида вычислительной сложности: Временная и ёмкостная.

Количество обращений к памяти характеризует временную сложность. Общий объём памяти, требуемой для решения задачи – характеристика ёмкостной сложности.

Алгоритмы бывают полиномиальные и экспоненциальные. Полиномиальные алгоритм – алгоритм, функция временной сложности которой является полиномом.

Функция сложности - качественные и количественные, характеристика алгоритма в зависимости от размера.

Алгоритм считается пригодным (приемлемым), если алгоритм – полиномиальный и степень полинома ≤4. Все остальные алгоритмы считаются экспоненциальными:

Асимптотическая оценка функции сложности. Пусть – реальная функция сложности, а – идеальная функция сложности. ;

Рассмотрим :



Ввод

Цикл

Цикл

Если

То

Всё если

Всё цикл

Всё цикл

Вывод

Шаг 1 Ввод |

Шаг 2 |1

Шаг 3 |1

Шаг 4 если то |4

Шаг 5 если , то , переход к 4 |3

Шаг 6 если , то , переход к 3 |3

Шаг 7 вывод |

*Если произвести подсчет шагов, с учетом их вложенности в циклы, то получим:*

*Трудно разрешимые задачи : классы P и NP*

*В конечном множестве машинных слов, которое занимает алгоритм имеется слово являющееся результатом его работы. Следовательно, поиск данного слова можно реализовать простым перебором, однако даже для малых слов на практике данный подход не применим, те задача алгоритмически решаема, а практически – нет.*

*Следоватльно можно сделать вывод, что для некоторых задач алгоритма лучшего чем перебор всех слов в алфавите – нет.*

*Приметами таких задач могут служить:*

* *Задача поиска ПНФ*
* *Задача коммивояжера.*

*Классы задач, рассматриваемые в теории вычислительной сложности:*

***Iй класс*** – существуют полиномиальные алгоритмы решения – **класс P (Polinomial)**

**Пример**: сортировка массива чисел.

***IIй класс*** –на данном этапе развития науки и математики нет полиномиального алгоритма, а постановка задачи влечет экспоненциальное время решения– **класс задач экспоненциальных по постановке**

*Пример: задача поиска всех подмножеств данного конечного множества.*

***IIIй класс*** –**алгоритмически неразрешимые проблемы**

*Пример: проблема остановки ТМ, задача распознавания выводимости в математической логике.*

***IVй класс*** –**задачи решаемые за полиномиальное время и проверяемые за полиномиальное время – класс NP (Non-deterministic Polinomial)**

*Пример: .*

*Для выделения данного класса задач строится недетерминированная машина Тьюринга (НМТ):*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *∞* | *0* | *1* | *0* | *0* | *0* | *1* | *0* | *0* | *∞* |

G2

G1

УУ

Оракул

*На ленте НМТ записана некоторая задача. Оракул за полиномиальное время находит (угадывает!!!) решение задачи, стирает ее условие на ленте и записывает полученный результат. Управление передается обычной машине Тьюринга в составе НМТ, которая выполняет проверку правильности решения за полиномиальное время.*

*Смысл задач класса NP – проверка решения задачи не легче ли чем ее непосредственное решение?*

*Множество всех задач разрешимых на НМТ за полиномиальное время -* **класс NP**

Задача Z принадлежит классу задач NP если:

* Задан конечный набор символов N
* Решение представляет собой конечный набор символов M
* f – полиномиальная функция

Задачи Clay Mathematic Institute США около 30 лет назад назначил премию в 1 миллион долларов за решение проблемы: эквивалентен ли класс NP классу P () или ?

Премия до сих пор не вручена. Проблема состоит в том, чтобы понять: если , то в классе NP существует хотя бы одна задача, не решаемая за полиномиальное время. С другой стороны, если , то все задачи класса NP полиномиальны по временной сложности, а это очень важно для эффективного решения большого числа задач. Большинство практически важных задач относится к классу NP.

Примеры задач класса NP:

* Пятнашки (игра) – предложена в 1878, Ноем Чепменом
* Возможность решения задачи раскраски графа 4мя цветами
* Нахождение максимальной клики в произвольном связном графе
* Задача покрытия неографа (нахождение множества вершин S таких, что каждое ребро графа инцидентно хотя бы одной вершине из S)
* Задача коммивояжера
* Задача построения минимального дерева Штейнера (соединение точек плоскости кратчайшей сетью)

***Vй класс*** – **NP полные задачи.**

Построение полиномиального алгоритма для одной задачи из класса NP полных влечет – за собой существование полиномиальных алгоритмов для задач подмножества исходной.

**NP полные задачи –** подмножество типовых задач класса NP. Полиномиальность одного алгоритма влечет полиномиальность целого класса алгоритмов.

Пусть есть 2 массовые задачи Z1 и Z2 такие, что они принадлежат классу NP. Говорят, что Z2 полиномиально сводима к Z1 , если для за полиномимиальное время можно найти решение задачи и при этом решение может быть за полиномиальное время преобразовано в решение Здесь и – индивидуальные задачи.

Данная сводимость в общем случае не тривиальна!

**Опр.** Массовая задача Z называется NP-полной, если любая задача из этого класса полиномиально сводима к Z.

Пример: исторически первая NP-полная задача – SAT (Satisfiability), в русскоязычной литературе задача «Выполнимость», состоящая в поиске набора аргументов булевой функции от n переменных так, чтобы составленная для данной функции КНФ была истинна.

Для того, чтобы доказать NP-полноту некоторой новой задачи необходимо:

* Установить ее принадлежность к классу NP
* Свести решениеэтой новой задачи к решению любой задачи из класса NP-полных.

На сегодня поиск сводимости задачи к классу NP-полных требует нетривиального эвристического подхода.