# Алгоритмы и структуры данных

Лекция 10. P versus NP.

(c) Глухих Михаил Игоревич, glukhikh@mail.ru

# Класс P

- Задачи, для решения которых существует алгоритм с полиномиальным временем работы
- Трудоёмкость $O(N^k)$
- Считаются имеющими «быстрое» решение

# Класс P

- Задачи, для решения которых существует алгоритм с полиномиальным временем работы
- Трудоёмкость $O(N^k)$
- Считаются имеющими «быстрое» решение

- Формально – на детерминированной машине Тьюринга (что это?)

# Машина Тьюринга =

- Бесконечная лента с символами
- Головка чтения-записи (может двигаться по ленте)
- Ограниченный набор состояний
- Символ + Состояние = Действие.
- Одно действие может включать в себя следующее
  - Записать символ
  - Переход в другое состояние
  - Смещение в соседнюю клетку

# Машина Тьюринга = демонстрация

- См. http://cmcmsu.info/1course/alg.schema.mt.htm
  - © ВМК МГУ

# Класс NP

- Задачи, для ПРОВЕРКИ решения которых существует алгоритм с полиномиальным временем работы
- Трудоёмкость ПРОВЕРКИ решения $O(N^P)$

# Класс NP

- Задачи, для ПРОВЕРКИ решения которых существует алгоритм с полиномиальным временем работы
- Трудоёмкость ПРОВЕРКИ решения $O(N^P)$
  - Эквивалентное определение: трудоёмкость РЕШЕНИЯ на НЕДЕТЕРМИНИРОВАННОЙ машине Тьюринга $O(N^P)$

# Класс NP

- Задачи, для ПРОВЕРКИ решения которых существует алгоритм с полиномиальным временем работы
- Трудоёмкость ПРОВЕРКИ решения $O(N^P)$
  - Эквивалентное определение: трудоёмкость РЕШЕНИЯ на НЕДЕТЕРМИНИРОВАННОЙ машине Тьюринга $O(N^P)$
  - Различие от детерминированной машины в том, что каждой комбинации символ + состояние может соответствовать более одного действия

# Класс NP

- Задачи, для ПРОВЕРКИ решения которых существует алгоритм с полиномиальным временем работы
- Трудоёмкость ПРОВЕРКИ решения $O(N^P)$

- Класс NP **включает в себя** класс P
- Т.е. любая задача из класса P входит в класс NP

# NP-полные задачи (класс NPC)

- ПОДКЛАСС класса NP (то есть, не то же самое)
- Задача из класса NP является NP-полной, если…

# NP-полные задачи (класс NPC)

- ПОДКЛАСС класса NP (то есть, не то же самое)
- Задача из класса NP является NP-полной, если…

- …к ней можно свести ЛЮБУЮ ДРУГУЮ NP-полную задачу за полиномиальное время – $O(N^k)$

# NP-полные задачи (класс NPC)

- ПОДКЛАСС класса NP (то есть, не то же самое)
- Задача из класса NP является NP-полной, если…
- …к ней можно свести ЛЮБУЮ ДРУГУЮ NP-полную задачу за полиномиальное время – $O(N^k)$
- Как правило, применяется к задачам с ответом Да / Нет (но обобщается и на другие задачи)

# NP-полные задачи (класс NPC)

- ПОДКЛАСС класса NP (то есть, не то же самое)
- Задача из класса NP является NP-полной, если…
- …к ней можно свести ЛЮБУЮ ДРУГУЮ NP-полную задачу за полиномиальное время – $O(N^k)$
- Как правило, применяется к задачам с ответом Да / Нет (но обобщается и на другие задачи)
- NB: для работы такого определения нужна хотя бы одна «базовая» NP-полная задача

# Базовая NP-полная задача

- SAT CNF (выполнимость булевых формул)

# Базовая NP-полная задача

- SAT CNF (выполнимость булевых формул)
- Верно ли, что существуют булевые значения x1, …, xN, такие, что заданная f(x1, …, xN) = 1?

# Базовая NP-полная задача

- SAT CNF (выполнимость булевых формул)
- Простейшая формулировка: есть логическая схема из элементов И, ИЛИ, НЕ, m входов, 1 выход

# Базовая NP-полная задача

- SAT CNF (выполнимость булевых формул)

- Простейшая формулировка: есть логическая схема из элементов И, ИЛИ, НЕ, m входов, 1 выход
- Существуют ли такие значения входов, при которых на выходе – 1?

# Базовая NP-полная задача

- SAT CNF (выполнимость булевых формул)
- Дана CNF(X) = T1(X)*T2(X)*…Tn(X), где X – вектор из m булевых переменных, а Ti(X) – термы-дизъюнкции
- Вопрос формулируется точно так же

# Варианты SAT: ДНФ

- F(x1, …, x5) = x1*!x2*x3 + x2*x3*!x4 + !x3*x4*x5
- x1, …, x5: F = 1?

# Варианты SAT: ДНФ

- F(x1, …, x5) = x1*!x2*x3 + x2*x3*!x4 + !x3*x4*x5
- x1, …, x5: F = 1?

- Решение очевидно существует: O(1)

# Варианты SAT: 2-КНФ

- F(x1, …, x5) = (x1 + !x2)*(x3 + x5)*(x4 + !x1)*(x2 + !x5)
- x1, …, x5: F = 1?

# Варианты SAT: 2-КНФ

- F(x1, …, x5) = (x1 + !x2)*(x3 + x5)*(x4 + !x1)*(x2 + !x5)
- x1, …, x5: F = 1?

- Решается за линейное время
  - **x1 = 0 => x2 = 0**, x2 = 1 => x1 = 1
  - x3 = 0 => x5 = 1, **x5 = 0 => x3 = 1**
  - x4 = 0 => x1 = 0, x1 = 1 => x4 = 1
  - **x2 = 0 => x5 = 0**, x5 = 1 => x2 = 1

  - x1 = 0 => x2 = 0 => x5 = 0 => x3 = 1 (x4 = 0/1)

# Другие NP-полные задачи

- Задача коммивояжёра
- Задача о сумме подмножеств
- Задача о ранце
- Задача о раскраске графа

# Задача коммивояжёра

- Поиск «самого дешёвого» пути через ВСЕ вершины на графе
- Решение?

# Задача коммивояжёра

- Поиск «самого дешёвого» пути через ВСЕ вершины на графе
- Решение?

- А) Перебор
- Б) Сокращённый перебор
- В) Эвристические методы

# Задача о сумме подмножеств

- Есть множество из целых чисел
- Найти в нём подмножество с нулевой суммой
- Решение?

# Задача о сумме подмножеств

- Есть множество из целых чисел
- Найти в нём подмножество с нулевой суммой
- Решение?

- А) Перебор всех вариантов

# Задача о сумме подмножеств

- Есть множество из N целых чисел
- Найти в нём подмножество с нулевой суммой
- Решение?

- А) Перебор всех вариантов
- Б) Динамическое программирование
  - Введём Q(K, S) = TRUE, если и только если существует подмножество первых K чисел, дающих в сумме S
  - S ограничено снизу суммой отрицательных чисел и сверху суммой положительных

# Задача о сумме подмножеств

- Есть множество X из N целых чисел
- Найти в нём подмножество с нулевой суммой
- Решение?

- А) Перебор всех вариантов
- Б) Динамическое программирование
  - Введём Q(K, S) = TRUE, если и только если существует подмножество первых K чисел, дающих в сумме S
  - S ограничено снизу суммой отрицательных чисел и сверху суммой положительных
  - Q(1, S) = (X1 == S)
  - Q(K, S) = Q(K – 1, S) OR (XK == S) OR Q(K – 1, S – XK)

# Задача о раскраске графа

- Необходимо раскрасить все вершины графа таким образом, чтобы все соседние вершины были разного цвета
- Хроматическое число = Минимальное количество цветов, для которых задача разрешима

# Задачи с недоказанной NP-полнотой

- Факторизация целых чисел (разложение на простые множители)

# P versus NP

- Фундаментальная проблема: совпадают ли классы задач P и NP?

# P versus NP

- Фундаментальная проблема: совпадают ли классы задач P и NP?
- Если НЕТ, то класс NPC не пересекается с классом P, то есть, NP-полные задачи нельзя решить за полиномиальное время

# P versus NP

- Фундаментальная проблема: совпадают ли классы задач P и NP?
- Если НЕТ, то класс NPC не пересекается с классом P, то есть, NP-полные задачи нельзя решить за полиномиальное время
- Если ДА, то P = NP = NPC, то есть, NP-полные задачи можно решить за полиномиальное время

# P versus NP

- Большинство учёных сейчас склоняется к мнению, что P <> NP

# P versus NP

- Большинство учёных сейчас склоняется к мнению, что P <> NP
- Предположим, тем не менее оказалось, что P = NP – какими будут последствия?

# P versus NP

- Большинство учёных сейчас склоняется к мнению, что P <> NP
- Предположим, тем не менее оказалось, что P = NP – какими будут последствия?
  - Есть приложения, предполагающие P <> NP, и они «сломаются»

# P versus NP

- Большинство учёных сейчас склоняется к мнению, что P <> NP
- Предположим, тем не менее оказалось, что P = NP – какими будут последствия?
  - Есть приложения, предполагающие P <> NP, и они «сломаются»
    - A) Криптосистемы с открытым ключом

# Далее

- Алгоритмы шифрации и дешифрации