

Вступ до теорії складності обчислень

Андрій Фесенко

01.09.2021

Що “складніше”?

- розв'язати рівняння $x^2 - 5x + 6 = 0$
або розв'язати рівняння
 $2^{78} \cdot x^2 - 5673971753579x + 111111011111 = 0$

Що “складніше”?

- розв'язати рівняння $x^2 - 5x + 6 = 0$
або розв'язати рівняння
 $2^{78} \cdot x^2 - 5673971753579x + 111111011111 = 0$
- розв'язати рівняння
 $2^{78} \cdot x^2 - 5673971753579x + 111111011111 = 0$
або розв'язати довільне квадратне рівняння

Що “складніше”?

- розв'язати рівняння $x^2 - 5x + 6 = 0$
або розв'язати рівняння
 $2^{78} \cdot x^2 - 5673971753579x + 111111011111 = 0$
- розв'язати рівняння
 $2^{78} \cdot x^2 - 5673971753579x + 111111011111 = 0$
або розв'язати довільне квадратне рівняння
- розв'язати довільне квадратне рівняння
або побудувати життєздатну колонію на Марсі

Теорія обчислень або **теорія алгоритмів** (англ. computability theory або recursion theory) є розділом математики та теоретичної інформатики, що досліджує задачі, які мають алгоритмічний розв'язок за певної моделі обчислень, ефективність та точність наявного розв'язку.

Теорія обчислень або **теорія алгоритмів** (англ. computability theory або recursion theory) є розділом математики та теоретичної інформатики, що досліджує задачі, які мають алгоритмічний розв'язок за певної моделі обчислень, ефективність та точність наявного розв'язку.

- **теорія автоматів**

Теорія обчислень або **теорія алгоритмів** (англ. computability theory або recursion theory) є розділом математики та теоретичної інформатики, що досліджує задачі, які мають алгоритмічний розв'язок за певної моделі обчислень, ефективність та точність наявного розв'язку.

- теорія автоматів
- теорія формальних мов

Теорія обчислень або **теорія алгоритмів** (англ. computability theory або recursion theory) є розділом математики та теоретичної інформатики, що досліджує задачі, які мають алгоритмічний розв'язок за певної моделі обчислень, ефективність та точність наявного розв'язку.

- теорія автоматів
- теорія формальних мов
- теорія обчислюваності (або теорія рекурсивних функцій)

Теорія обчислень або **теорія алгоритмів** (англ. computability theory або recursion theory) є розділом математики та теоретичної інформатики, що досліджує задачі, які мають алгоритмічний розв'язок за певної моделі обчислень, ефективність та точність наявного розв'язку.

- теорія автоматів
- теорія формальних мов
- теорія обчислюваності (або теорія рекурсивних функцій)
- **теорія складності обчислень**

Теорія обчислень або **теорія алгоритмів** (англ. computability theory або recursion theory) є розділом математики та теоретичної інформатики, що досліджує задачі, які мають алгоритмічний розв'язок за певної моделі обчислень, ефективність та точність наявного розв'язку.

- **теорія автоматів**
- **теорія формальних мов**
- **теорія обчислюваності (або теорія рекурсивних функцій)**
- **теорія складності обчислень**
 - **аналіз алгоритмів**

- використання формальних моделей обчислень (теорія обчислюваності)

Предмет теорії складності обчислень

- використання формальних моделей обчислень (теорія обчислюваності)
- визначення кількості необхідних ресурсів для обчислень

Предмет теорії складності обчислень

- використання формальних моделей обчислень (теорія обчислюваності)
- визначення кількості необхідних ресурсів для обчислень
- дослідження складності алгоритмів (аналіз алгоритмів)

Предмет теорії складності обчислень

- використання формальних моделей обчислень (теорія обчислюваності)
- визначення кількості необхідних ресурсів для обчислень
- дослідження складності алгоритмів (аналіз алгоритмів)
- побудова класифікації обчислювальних задач у відповідності до кількості використовуваних ресурсів

Предмет теорії складності обчислень

- використання формальних моделей обчислень (теорія обчислюваності)
- визначення кількості необхідних ресурсів для обчислень
- дослідження складності алгоритмів (аналіз алгоритмів)
- побудова класифікації обчислювальних задач у відповідності до кількості використовуваних ресурсів
- створення методів класифікації задач

Предмет теорії складності обчислень

- використання формальних моделей обчислень (теорія обчислюваності)
- визначення кількості необхідних ресурсів для обчислень
- дослідження складності алгоритмів (аналіз алгоритмів)
- побудова класифікації обчислювальних задач у відповідності до кількості використовуваних ресурсів
- створення методів класифікації задач
- використання класифікації задач для аналізу методів розв'язку

\mathcal{F}_1 — множина всіх часткових функцій виду $\mathbb{N}_0 \rightarrow \mathbb{N}_0$

Об'єкт дослідження теорії обчислень

\mathcal{F}_1 — множина всіх часткових функцій виду $\mathbb{N}_0 \rightharpoonup \mathbb{N}_0$

сучасний підхід:

використання формальних мов

множина всіх часткових функцій виду $\{0,1\}^* \rightharpoonup \{0,1\}^*$

множина всіх функцій виду $\{0,1\}^* \rightarrow \{0,1\}$

Об'єкт дослідження теорії обчислень

\mathcal{F}_1 — множина всіх часткових функцій виду $\mathbb{N}_0 \rightharpoonup \mathbb{N}_0$

сучасний підхід:

використання формальних мов

множина всіх часткових функцій виду $\{0, 1\}^* \rightharpoonup \{0, 1\}^*$

множина всіх функцій виду $\{0, 1\}^* \rightarrow \{0, 1\}$

Приклад

Число $12_{10} = 1100_2$, але $001100_2 = 01100_2 = 12_{10}$

- $001100_2 \mapsto 1001100_2 = 76_{10}$ і $01100_2 \mapsto 101100_2 = 44_{10}$

Модель обчислень є формалізованою моделлю, яка описує як обчислюється результат відображення за заданим вхідним значенням.

З точки зору теорії обчислюваності — підмножина множини всіх часткових функцій виду $\{0, 1\}^* \rightarrow \{0, 1\}^*$

- послідовні (англ. sequential)

- послідовні (англ. sequential)
 - скінченні автомати

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - **автомати з магазинною пам'яттю**

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)
 - частково рекурсивні функції

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)
 - частково рекурсивні функції
 - λ -числення

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)
 - частково рекурсивні функції
 - λ -числення
- паралельної обробки даних (англ. concurrent)

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)
 - частково рекурсивні функції
 - λ -числення
- паралельної обробки даних (англ. concurrent)
 - модель акторів

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)
 - частково рекурсивні функції
 - λ -числення
- паралельної обробки даних (англ. concurrent)
 - модель акторів
 - клітинні автомати

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)
 - частково рекурсивні функції
 - λ -числення
- паралельної обробки даних (англ. concurrent)
 - модель акторів
 - клітинні автомати
 - мережі Петрі

Модель обчислень

- послідовні (англ. sequential)
 - скінченні автомати
 - машини Поста
 - автомати з магазинною пам'яттю
 - машини з довільним доступом до пам'яті або RAM-машини
 - машини Тюрінга
- функціональні (англ. functional)
 - абстрактні системи переписування (нормальні алгоритми Маркова)
 - частково рекурсивні функції
 - λ -числення
- паралельної обробки даних (англ. concurrent)
 - модель акторів
 - клітинні автомати
 - мережі Петрі
 - модель елементарних вентилів

Обов'язкові фундаментальні властивості алгоритму:

- **скінченність** (або **об'єктивність**)

Обов'язкові фундаментальні властивості алгоритму:

- **скінченність** (або **об'єктивність**)
- **дискретність**

Обов'язкові фундаментальні властивості алгоритму:

- скінченність (або **об'єктивність**)
- дискретність
- **детермінованість**

Властивості алгоритму

Обов'язкові фундаментальні властивості алгоритму:

- **скінченність (або об'єктивність)**
- **дискретність**
- **детермінованість**

Необов'язкові властивості алгоритму:

- **масовість алгоритму**

Властивості алгоритму

Обов'язкові фундаментальні властивості алгоритму:

- **скінченність (або об'єктивність)**
- **дискретність**
- **детермінованість**

Необов'язкові властивості алгоритму:

- **масовість алгоритму**
- **правильність**

Властивості алгоритму

Обов'язкові фундаментальні властивості алгоритму:

- **скінченність (або об'єктивність)**
- **дискретність**
- **детермінованість**

Необов'язкові властивості алгоритму:

- **масовість алгоритму**
- **правильність**
- **завершуваність**

Ресурси алгоритму

- **час виконання**

Ресурси алгоритму

- час виконання
- пам'ять

Ресурси алгоритму

- час виконання
- пам'ять
- енергія

Ресурси алгоритму

- час виконання
- пам'ять
- енергія
- кількість елементарних вентилів

Ресурси алгоритму

- час виконання
- пам'ять
- енергія
- кількість елементарних вентилів
- кількість обмінів повідомленнями

Ресурси алгоритму

- час виконання
- пам'ять
- енергія
- кількість елементарних вентилів
- кількість обмінів повідомленнями
- кількість процесорів (розпаралелювання)

Ресурси алгоритму

- час виконання
- пам'ять
- енергія
- кількість елементарних вентилів
- кількість обмінів повідомленнями
- кількість процесорів (розпаралелювання)
- розподіленість

Обчислювальна задача

Означення

Математичний об'єкт, який представляється множиною питань, що можуть бути вирішені за допомогою алгоритмічних методів в певній моделі обчислень. Нескінченна множина **екземплярів** задач з, можливо порожньою, множиною розв'язків для кожного екземпляра.

Обчислювальна задача

Означення

Математичний об'єкт, який представляється множиною питань, що можуть бути вирішені за допомогою алгоритмічних методів в певній моделі обчислень. Нескінченна множина **екземплярів** задач з, можливо порожньою, множиною розв'язків для кожного екземпляра.

масова задача **vs** індивідуальна задача

Обчислювальна задача

Означення

Математичний об'єкт, який представляється множиною питань, що можуть бути вирішені за допомогою алгоритмічних методів в певній моделі обчислень. Нескінченна множина **екземплярів** задач з, можливо порожньою, множиною розв'язків для кожного екземпляра.

масова задача **vs** індивідуальна задача

Види обчислювальних задач

- задачі пошуку (бінарне відношення)

Обчислювальна задача

Означення

Математичний об'єкт, який представляється множиною питань, що можуть бути вирішені за допомогою алгоритмічних методів в певній моделі обчислень. Нескінченна множина **екземплярів** задач з, можливо порожньою, множиною розв'язків для кожного екземпляра.

масова задача **vs** індивідуальна задача

Види обчислювальних задач

- задачі пошуку (бінарне відношення)
- задачі розпізнавання (відповідь – 'так' або 'ні', $Y_n \subseteq D_n$)

Обчислювальна задача

Означення

Математичний об'єкт, який представляється множиною питань, що можуть бути вирішені за допомогою алгоритмічних методів в певній моделі обчислень. Нескінченна множина **екземплярів** задач з, можливо порожньою, множиною розв'язків для кожного екземпляра.

масова задача **vs** індивідуальна задача

Види обчислювальних задач

- задачі пошуку (бінарне відношення)
- задачі розпізнавання (відповідь – 'так' або 'ні', $Y_n \subseteq D_n$)
- задачі оптимізації (пошук найкращого розв'язку)

Обчислювальна задача

Означення

Математичний об'єкт, який представляється множиною питань, що можуть бути вирішені за допомогою алгоритмічних методів в певній моделі обчислень. Нескінченна множина **екземплярів** задач з, можливо порожньою, множиною розв'язків для кожного екземпляра.

масова задача **vs** індивідуальна задача

Види обчислювальних задач

- задачі пошуку (бінарне відношення)
- задачі розпізнавання (відповідь – 'так' або 'ні', $Y_n \subseteq D_n$)
- задачі оптимізації (пошук найкращого розв'язку)
- задачі підрахунку (кількість розв'язків)

Обчислювальна задача

Означення

Математичний об'єкт, який представляється множиною питань, що можуть бути вирішені за допомогою алгоритмічних методів в певній моделі обчислень. Нескінченна множина **екземплярів** задач з, можливо порожньою, множиною розв'язків для кожного екземпляра.

масова задача **vs** індивідуальна задача

Види обчислювальних задач

- задачі пошуку (бінарне відношення)
- задачі розпізнавання (відповідь – 'так' або 'ні', $Y_n \subseteq D_n$)
- задачі оптимізації (пошук найкращого розв'язку)
- задачі підрахунку (кількість розв'язків)
- функціональні задачі

Типи складності за індивідуальними задачами

числовий параметр — **розмір індивідуальної задачі**

Типи складності за індивідуальними задачами

числовий параметр — **розмір індивідуальної задачі**

- **складність в найгіршому випадку**

Типи складності за індивідуальними задачами

числовий параметр — **розмір індивідуальної задачі**

- складність в найгіршому випадку
- складність в середньому

Типи складності за індивідуальними задачами

числовий параметр — **розмір індивідуальної задачі**

- складність в найгіршому випадку
- складність в середньому
- складність в найкращому випадку

Типи складності за індивідуальними задачами

числовий параметр — **розмір індивідуальної задачі**

- складність в найгіршому випадку
- складність в середньому
- складність в найкращому випадку

Складність задачі визначається складністю найкращого розв'язку у найгіршому випадку на асимптотиці