

СКЛАДНІСНІ КЛАСИ ЗАДАЧ

Вступ

- **На початку 1960-х р.**, у зв'язку з початком широкого використання обчислювальної техніки (ОТ) для вирішення практичних завдань, **виникло питання про межі практичної застосовності конкретного алгоритму розв'язання задачі у сенсі обмежень на її розмірність. Які завдання можуть бути вирішені на ЕОМ за реальний час?**
- **Відповідь на це питання була дана у працях Кобмена (Alan Cobham, 1964) та Едмондса (Jack Edmonds, 1965), де були введені класи складності задач.**
- **У теорії алгоритмів (ТА) класами складності називаються множини обчислювальних задач, приблизно однакових за складністю обчислення.**
- **Говорячи вужче, класи складності - це множини предикатів (функцій, які отримують на вхід слово і повертають відповідь 0 або 1), що використовують для**

Вступ

- У рамках класичної теорії здійснюється класифікація завдань за класами складності:
 - P-складні;
 - NP-складні;
 - експоненціально складні;
 - тощо.
- Кожен клас складності (у вузькому сенсі) визначається як множина предикатів, що мають деякі властивості.

Означення. *Класом складності X називається множина предикатів $P(x)$, що обчислюються на машинах Тюрінга (МТ) і використовуються для обчислення*

$O(f(n))$ ресурсу, де n - довжина слова x .

Вступ

- **Як ресурси зазвичай береться час обчислення** (кількість робочих тактів МТ) **або робоча зона** (кількість використаних комірок на рядку під час роботи).
- Мови, які розпізнаються предикатами з деякого класу (тобто множини слів, на яких предикат повертає 1), також називаються мовами, що належать до того самого класу. Класи прийнято позначати прописними літерами. Доповнення до класу C (тобто клас мов, доповнення яких належать C) позначається co-C.
- **Для кожного класу є категорія задач, які є «найскладнішими».** Це означає, що будь-яка задача з класу зводиться до такої задачі, і до того ж сама задача міститься у класі. **Такі задачі називають повними задачами для даного класу.**
- Найвідомішими є **NP-повні задачі.**

Тайвідоміші класи складності задач

- **1. КЛАС P (ЗАДАЧІ З ПОЛІНОМІАЛЬНОЮ СКЛАДНІСТЮ).**
- **Означення.** Алгоритм ототожнюється з детермінованою МТ, яка обчислює відповідь при даному на вхідний рядок слові з вхідного алфавіту Σ . Час роботи алгоритму $T_M(x)$ при фіксованому вхідному слові x визначається кількістю робочих тактів МТ від початку до зупинки МТ. $C_M(n) = \max_{x: |x|=n} T_M(x)$
- **Означення.** Якщо для функції f існує МТ M така, що $C_M(n) < n^c$ для деякого числа c і досить великих n , то кажуть, що вона належить класу P , або поліноміальна за часом.
- **Задача називається поліноміальною (належить до класу P), якщо існують константа k та алгоритм, що**

ГЛАВІ ВІДОМІШІ КЛАСИ СКЛАДНОСТІ

Задачі

- **Згідно з тезою Черча-Тюрінга** будь-який мислимий алгоритм можна реалізувати на МТ. Для будь-якої мови програмування (МП) можна визначити клас P подібним чином (замінивши у визначенні МТ на реалізацію МП). Якщо компілятор мови, на якому реалізований алгоритм, уповільнює виконання алгоритму поліноміально (тобто час виконання алгоритму на МТ менший за деякий многочлен від часу виконання його на МП), то визначення класів P для цієї МП і для МТ збігаються.
- **Задачі класу P - задачі, що розв'язувані за реальний час.**
- **Основні переваги алгоритмів класу P :**
 - для більшості задач класу P константа k менше 6;
 - клас P інваріантний за моделлю обчислень (для широкого класу моделей);
 - клас P має властивість природної замкненості (сума або добуток поліномів є поліном).
- **Задачі класу P є уточненням визначення «практично розв'язуваної» задачі.**
- **Приклади завдань класу P : цілочислове додавання, множення, ділення, одержання залишку від ділення, множення матриць, з'ясування зв'язності графів та інші.**

задач

- **2. КЛАС NP (ПОЛІНОМІАЛЬНОЇ ПЕРЕВІРКИ).**
- *Клас NP складається із задач, розв'язання яких можна перевірити протягом поліноміального часу.* Тобто, що якщо ми одержимо якимось чином розв'язок деякої задачі цього класу, то протягом часу, який буде поліноміально залежати від розміру вхідних даних задачі, можна перевірити коректність такого розв'язку.
- **Означення.** $\forall D \in D_{A'}$, $|D|=n$ поставимо у відповідність сертифікат $S \in S_{A'}$, такий, що $|S| = O(n^l)$, та алгоритм $A_S = A_S(D, S)$, такий, що видає «1», якщо розв'язок правильний, і «0» - якщо неправильний. Тоді задача належить до класу NP, якщо $F(A_S) = O(n^m)$.
- Еквівалентне визначення можна отримати, використовуючи поняття недетермінованої МТ (тобто такої МТ, у програмі якої можуть існувати різні рядки з

Найвідоміші класи складності

задач

- **Основна відмінність класів P і NP:**

- **до класу P** належить задачі, які можуть бути розв'язані за час, що поліноміально залежить від об'єму початкових даних, за допомогою детермінованої обчислювальної машини (наприклад, МТ),

- **до класу NP** належить задачі, які можуть бути розв'язані за поліноміально виражений час за допомогою недетермінованої обчислювальної машини, тобто машини, наступний стан якої не завжди однозначно визначається попередніми. Роботу такої машини можна представити як процес, що розгалужується на кожній неоднозначності: задача вважається розв'язаною, якщо хоча б одна гілка процесу прийшла до відповіді. Оскільки кожна детермінована МТ може розглядатись як недетермінована, але без вибору варіантів кроків, то клас P міститься у класі NP ($P \subseteq NP$)

Поняття МТ, ДМТ, НМТ

- До складу МТ входить необмежена в обидві сторони стрічка, розділена на комірки, і керуючий пристрій - голівка запису-читання (далі голівка), здатна перебувати в одному з множини станів. Число можливих станів керуючого пристрою є кінцевим і точно заданим.
- Керуючий пристрій може переміщатися вліво і вправо по стрічці, читати і записувати у комірці символи деякого кінцевого алфавіту. Виділяється особливий порожній символ, що заповнює усі клітини стрічки, крім тих з них (кінцевого числа), на яких записані вхідні дані.
- Керуючий пристрій працює згідно з правилами переходу, які представляють алгоритм, реалізований цією МТ. Кожне правило переходу наказує МТ, залежно від поточного стану і спостережуваного у поточній комірці символу, записати в цю комірку новий символ, перейти у новий стан і переміститися на одну комірку вліво або вправо. Деякі стани машини Тьюринга можуть бути позначені як термінальні, і перехід в будь-який з них означає кінець роботи, зупинку алгоритму.

Поняття МТ, ДМТ, НМТ (продовження)

- Конкретна МТ задається перерахуванням елементів множини букв алфавіту A , множини станів Q і набором правил, за якими працює МТ. Вони мають вигляд:

$$q_i a_j \rightarrow q_{i_1} a_{j_1} d_k$$

якщо головка знаходиться у стані q_i , а у комірці, що спостерігається записана буква a_j , то голівка переходить у стан q_{i_1} , у комірку замість a_j записується a_{j_1} , голівка робить рух d_k , який має 3 варіанти: на комірку вліво (L), вправо (R), або залишається на місці (N).

- **Для кожної можливої конфігурації $\langle q_i, a_j \rangle$ існує рівно одне правило (для недетермінованої МТ може бути більше правил).** Правил немає тільки для заключного стану, потрапивши у яке, МТ зупиняється. Крім того, необхідно вказати кінцевий і початковий стани, початкову конфігурацію на стрічці і розташування голівки МТ.

Поняття МТ, ДМТ, НМТ (продовження)

- МТ є детермінованою, якщо кожній комбінації стану і стрічкового символу у таблиці відповідає не більше одного правила. Якщо існує пара «стрічковий символ - стан», для якої існує 2 і більше команд, така МТ називається недетермінованою.

Іншими словами:

- Детермінована МТ (ДМТ) має функцію переходу, яка по комбінації поточного стану і символу на стрічці визначає 3 речі:
 - символ, що буде записаний на стрічці;
 - напрямок зміщення голівки по стрічці;
 - новий стан скінченного автомату.

Поняття МТ, ДМТ, НМТ (продовження)

- Для недетермінованої МТ (НМТ), комбінація поточного стану автомата і символу на стрічці може допускати кілька переходів. Як НМТ «дізнається», який з можливих шляхів приведе в потрібний стан? Є 2 способи це уявити:
 - можна вважати, що НМТ завжди вибирає перехід, який у кінцевому підсумку приводить до потрібного стану, якщо такий перехід взагалі є;
 - можна уявити, що у разі неоднозначності переходу (поточна комбінація стану і символу на стрічці допускає кілька переходів) НМТ ділиться на кілька НМТ, кожна з яких діє за одним з можливих переходів.
- Тобто на відміну від ДМТ, яка має єдиний «шлях обчислень», НМТ має «дерево обчислень» (у загальному випадку - експоненціальне число

Найвідоміші класи складності

задач

- Оскільки клас P міститься у класі NP , належність того або іншого завдання до класу NP часто відображає наше поточне уявлення про способи розв'язання даної задачі й носить неостаточний характер.
- До задач класу складності NP належать, наприклад, задачі: розв'язність логічного виразу, три-кольорове розфарбування графу, побудова кліки з вершин на неографі, задача покриття множин та інші.
- У загальному випадку немає підстав вважати, що для тієї або іншої NP -задачі не може бути знайдений P -розв'язок. Питання про можливу еквівалентність класів P і NP (тобто про можливість знаходження P -розв'язку для будь-якої NP -задачі) вважається багатьма одним з основних питань сучасної теорії складності алгоритмів. Сама постановка питання про еквівалентність класів P і NP можлива завдяки введенню поняття NP -повних задач.

Найвідоміші класи складності

задач

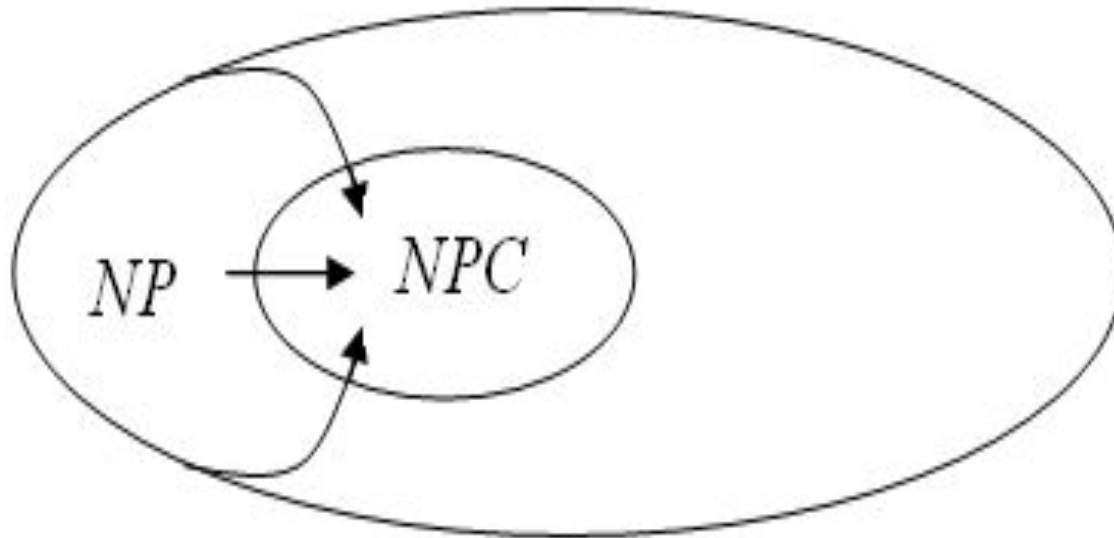
- **3. КЛАС NPC** (*NP - ПОВНІ ЗАДАЧІ*)
- Неформально задача належить класу NPC, якщо вона належить класу NP і є такою самою «складною», як і будь-яка задача з класу NP.
- NP-повні задачі складають підмножину NP-задач і відрізняються тією властивістю, що всі NP-задачі можуть бути тим або іншим способом зведені до них. З цього виходить, що якщо для NP-повної задачі буде знайдений P-розв'язок, то P-розв'язок буде знайдений для усіх задач класу NP.
- Поняття NP-повноти було введено на початку 1970-х років і ґрунтується на понятті зведення однієї задачі до іншої.

задач

- Зведення може бути представлене у такий спосіб: якщо ми маємо задачу 1 та алгоритм, що розв'язує цю задачу, тобто видає правильну відповідь для всіх конкретних завдань, що становлять задачу, а для задачі 2 алгоритм розв'язання невідомий, але якщо ми можемо переформулювати (звести) задачу 2 у термінах задачі 1, то ми розв'язуємо задачу 2.
- Таким чином, якщо задача 1 задана множиною конкретних завдань D_{A1} , а задача 2 – множиною D_{A2} , й існує функція f_s (алгоритм), що зводить конкретну постановку задачі 2 (d_2) до конкретної постановки задачі 1 (d_1): $f_s(d_2 \in D_{A2}) = d_1 \in D_{A1}$, то задача 2 зводиться до задачі 1.
- Якщо при цьому $F(f_s) = O(n^k)$, тобто алгоритм зведення належить класу P , то говорять, що задача 1 поліноміально зводиться до задачі 2

Найвідоміші класи складності задач

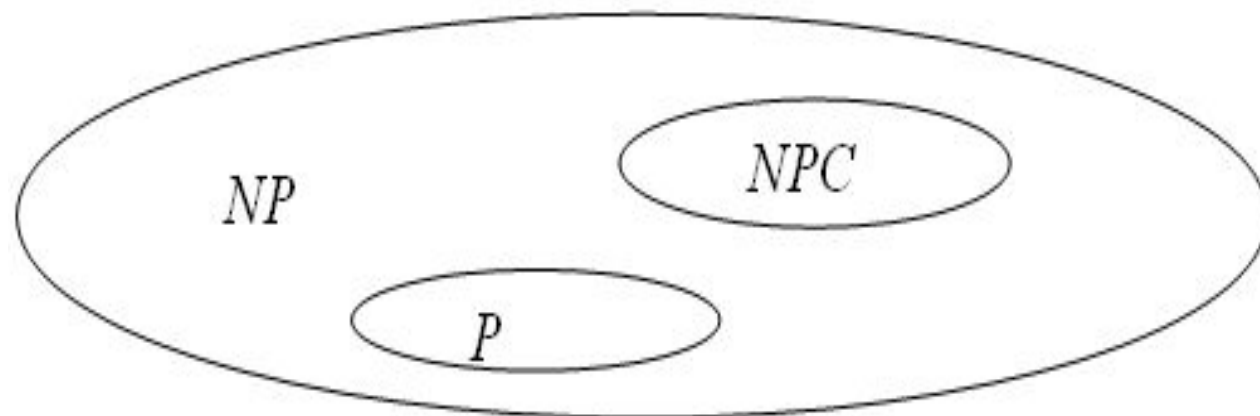
- **Означення.** Задача належить до класу NPC, якщо виконуються такі дві умови: 1) задача повинна належати до класу NP ($L \in NP$); 2) до задачі поліноміально повинні зводитися всі задачі з класу NP ($L_x \leq p$, для кожного $L_x \in NP$).
- Виконання цього означення показано на рис.:



Найвідоміші класи складності

задач

- **Теорема 1.** Якщо існує задача, що належить до класу NPC, для якої існує поліноміальний алгоритм розв'язання ($F=O(n^k)$), то клас P збігається з класом NP, тобто $P=NP$.
- Схема доведення буде складатися зі зведення будь-якої задачі з NP до даної задачі з класу NPC з поліноміальною трудомісткістю й розв'язання цієї задачі за поліноміальний час (за умовою теореми).
- Натепер доведено існування сотень NPC задач, але для жодної з них поки не вдалося знайти поліноміального алгоритму розв'язання. У цей час дослідники припускають співвідношення класів, що наведено на рис., де $P \neq NP$ і задачі з класу NPC не можуть бути розв'язані (сьогодні) з поліноміальною трудомісткістю.



Найвідоміші класи складності

задач

- Оскільки метод зведення базується на тому, що для деякої задачі відома її NP – повнота, то для доведення NP – повноти різних задач, нам знадобиться «перша» NP – повна задача. Як таку використовуємо задачу на здійсненність, у якій задана булева комбінаційна схема, що складається з логічних елементів (або задано КНФ – кон'юнктивну нормальну форму). У задачі запитується, чи існує набір вхідних булевих величин, для яких на виході буде видане значення 1.
- Прикладами NP-повних задач є, крім задачі про кон'юнктивну форму, задача комівояжера, розфарбування графа, задача про гамільтонів цикл у графі та інші.

Інші класи складності задач

- Дослідження складності алгоритмів дозволили по-новому подивитися на розв'язання багатьох класичних математичних задач і знайти для ряду таких задач (множення многочленів і матриць, розв'язання систем лінійних рівнянь та ін.), розв'язання, що вимагають менше ресурсів, ніж традиційні.
- **Прості класи складності визначаються такими факторами:**
- **типом обчислювальної задачі:** найбільш часто використовуються задачі прийняття рішень.
- **моделлю обчислень:** найбільш поширеною моделлю обчислень є детермінована МТ, але багато класів складності визначають на основі недетермінованої МТ, логічних схем, квантової МТ, монотонних схем і т.д.
- **ресурсами, які мають межі:** вказується як "поліноміальний час", "логарифмічний простір", "стала глибина" тощо.
- Усі класи складності знаходяться в ієрархічному відношенні: одні містять у собі інші. Однак про більшість включень невідомо, чи є вони строгими. Одна з найвідоміших відкритих проблем у цій сфері – рівність класів P і NP. Натепер найпоширенішою є гіпотеза про невиродженість ієрархії (тобто усі класи різні).