

Доведено, що наведені вище методи мають такі властивості:

- узгодженість методу з p -м порядком точності формули Тейлора на кожній ітерації m ($m=0,1, 2, \dots$);
- А-стійкість (для випадку а) або L-стійкість (для випадку б) при розв'язанні модельного рівняння

$$Y' = -\lambda Y, \quad \operatorname{Re}(\lambda) > 0;$$

- збіжність до розв'язку $\bar{Y}_{n+1}^{[p]}$;
- не вимагають обернення матриць Якобі або розв'язання систем алгебраїчних рівнянь;
- можуть бути застосовані для розв'язання і нежорстких задач;
- можуть бути реалізовані на багатопроцесорних комплексах, оскільки допускають розпаралелювання алгоритмів.

Розглянуті методи реалізовані в підпрограмі, яка пройшла успішні випробування на тестових прикладах, які при відповідних вхідних даних описують рівняння з певним типом властивостей. Аналіз результатів тестування показав, що запропоновані методи є достатньо ефективними з точки зору обчислювальних затрат і точності розв'язку.

УДК 621.3

А.О. Мельник, М.В. Черкаський

Національний університет "Львівська політехніка",
кафедра ЕОМ

ТЕОРІЯ АЛГОРИТМІВ І МЕТОДИ ОБЧИСЛЕНЬ: НОВИЙ КУРС

© Мельник А.О., Черкаський М.В., 2001

Порівнюються дві програми з теорії алгоритмів і методів обчислень для базового напрямку "Комп'ютерна інженерія". Перша – один з початкових варіантів програми-рекомендації для університетів США, друга – апробована протягом трьох років програма Національного університету "Львівська політехніка". Показані переваги і недоліки обох програм та запропоновано новий варіант програми. Обґрунтовано використання програмно-апаратної моделі алгоритму, наближеної до реальних комп'ютерних засобів.

The algorithm theory and computational methods programs for base direction "Computer engineering" are compared. The original, recommended for the Universities in USA and the one proposed for National University "Lviv Polytechnic" and approbated three years. Advantages and disadvantages of both are discussed and new variant of the program is proposed. The benefit of the software/hardware model algorithm approach, approximate to real computing systems, is substantiated.

Вступ. У березні 2001 року у Львові відбулось засідання НМК Міністерства освіти і науки України з базового напрямку "Комп'ютерна інженерія", на якій обговорювались зміни в навчальних планах бакалаврату. На засіданні з доповіддю про сучасний стан фаху виступив завідувач кафедри СКС НТУУ КПІ професор Тарасенко В.П., який ознайомив із сучасним змістом програм навчання фаху в університетах за кордоном. У результаті обговорення були скоректовані навчальні плани базового напрямку "Комп'ютерна інженерія". На основі цих планів зароз формуються нові навчальні програми дисциплін.

У статті розглядається лише одна дисципліна – “Теорія алгоритмів і методи обчислень”, яка включена до нового навчального плану. На кафедрі ЕОМ Львівської політехніки курс “Основи теорії алгоритмів” викладається три роки, тому доцільно з метою покращання програми, що розробляється, провести порівняльний аналіз матеріалів, про які доповідав професор Тарасенко В.П. і програму кафедри ЕОМ. Такий порівняльний аналіз проведено в даній статті для двох програм, присвячених дисципліні “Теорія алгоритмів і методи обчислень” – попереднього варіанту програми-рекомендації для університетів США і програми Львівської політехніки (додатки 1 і 2).

Аналіз програми-рекомендації. Ця програма містить десять розділів, причому перші п’ять виділені як ядро побудови робочих програм. Об’єм цих розділів становить 31 годину. Решта п’ять розділів – це доповнення до першої частини або розділи, пов’язані із спеціалізацією навчання згідно з науковим напрямком кафедри.

Перший розділ (ядро – 4 години) присвячений основам обчислення часової і частково місткісної складності неформальних алгоритмів. За змістом до першого розділу належить також шостий розділ, який описує N - і NP -класи складності. До переліку загальних понять доцільно включити опис параметрів і властивостей алгоритмів, а також розглянути історію уточнення тлумачення поняття алгоритму, його характеристик, які відсутні в програмі-рекомендації.

Другий розділ (ядро – 6 годин), названий “Алгоритмічні стратегії”, описує способи (всього сім) конструювання ефективних неформальних алгоритмів. Не названі способи: еквівалентні перетворення, використання попередніх обчислень, зміна правила початку. Не уточнений пункт “Поділяй і пануй”, до якого можна віднести: розділення множини вхідних і вихідних даних на підмножини, а також розподіл задач на підзадачі.

Третій розділ “Фундаментальні комп’ютерні алгоритми”(ядро – 12 годин) дає перелік алгоритмів з послідовним виконанням операцій. Цей розділ присвячений, головним чином, демонстрації прикладів ефективних неформальних алгоритмів пошуку і сортування.

Четвертий розділ (ядро – 3 години) підкреслює особливості реалізації розподілених алгоритмів.

Теоретичні основи алгоритмів розглядаються в п’ятому розділі “Основи комп’ютерної теорії” (ядро – 6 годин) і сьомому “Теорія автоматів”. П’ятий розділ стосується машин з кінцевим станом, граматик безконтекстних мов, необчислювальних функцій. Сьомий розділ є найбільш об’ємним. Він містить пункти, пов’язані з детермінованими і недетермінованими автоматами, машинами Тьюрінга, властивостями граматик безконтекстних мов, ієрархії Хомського.

Сучасним актуальним алгоритмічним системам присвячений восьмий розділ, який містить синтез алгоритмів у реальному часі, амортизаційний аналіз, імовірнісні алгоритми, динамічне програмування, комбінаторну оптимізацію.

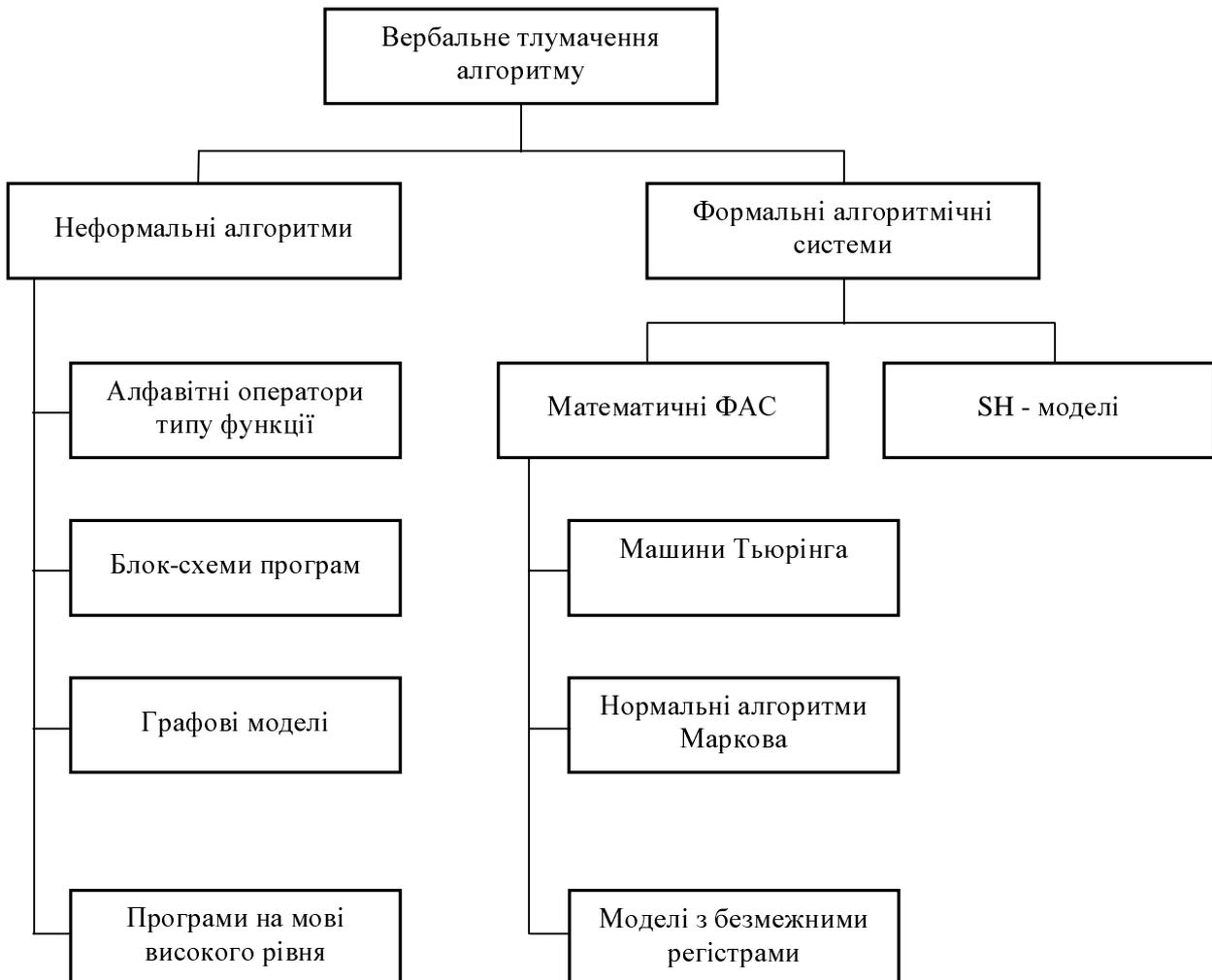
Дев’ятий і десятий розділи фіксують актуальні напрямки дослідження задач – таких, як криптографія і оброблення зображень, що розв’язуються проблемно-орієнтованими і спеціалізованими комп’ютерними системами.

Отже, американська програма перераховує багато розділів, що належать до області неформальних алгоритмів. Вона містить достатньо матеріалів для побудови курсів з алгоритмів і часової складності з врахуванням спрямування кафедр на розв’язання тих чи інших алгоритмічних проблем.

Разом з тим структура програми не підпорядкована цілісній концепції, що характеризує сучасний стан теорії і практики алгоритмів і алгоритмічних систем. Відсутність розділів з історії уточнення поняття алгоритму, опису властивостей і параметрів алгоритму, опису формальних і неформальних алгоритмів затруднює цілісне розуміння окремих розділів.

Розділ, що стосується синтезу ефективних алгоритмів, присутній по замовчуванню. Ефективність алгоритмів трактується лише за часовою складністю, інші характеристики складності не розглядаються. Відсутні також розділи, що стосуються алгоритмів, які реалізуються апаратними або апаратно-програмними засобами.

Програма Львівської політехніки. Кафедра ЕОМ має досвід викладання дисципліни “Основи теорії алгоритмів”, яка охоплює також основи метричної теорії складності, для студентів базового напрямку “Комп’ютерна інженерія”(див. додаток 2). Концепція побудови дисципліни базується на класифікації алгоритмів (див. рис.).



Вербальне тлумачення алгоритму. В першому розділі розкривається історія уточнення поняття алгоритму, основні визначення, перелік властивостей і параметрів, обговорюються проблема розв’язності і метрика алгоритмів, що становлять основний предмет дослідження теорії алгоритмів. Дається визначення задачі, підкреслюється неподільний зв’язок задачі і алгоритму, обговорюються їх спільні структурні елементи. При опису властивостей алгоритму підкреслюється неможливість математично точного визначення елементарності кроку. Бажання уточнити поняття елементарності привело до появи значної кількості моделей алгоритмів. Залежно від того, чи є крок алгоритму інтегрованим чи ні, алгоритми розподіляються на два великі класи – неформальні і формальні.

Неформальні алгоритми. Під неформальними розуміють алгоритми, для яких правило безпосереднього перероблення задано переліком взаємозв’язаних операцій, але не вказано,

як передбачається виконувати ці операції. Кроки неформальних алгоритмів мають інтегральний характер. Прикладами моделей неформальних алгоритмів є алфавітний оператор виду функцій, послідовність речень, яка використовує символи деякого абстрактного алфавіту, блок-схема програми, графічні моделі, послідовність операторів на алгоритмічних мовах високого рівня та інші. Алфавітні оператори є об'єктом перетворення з метою одержання еквівалентних їм алфавітних операторів з мінімізованою часовою складністю. Способами конструювання ефективних неформальних алгоритмів є: еквівалентні перетворення, апроксимація, використання попередніх обчислень, зокрема табличний спосіб, різні форми розбиття множин початкових даних та проміжних результатів на підмножини, розбиття множини можливих результатів на підмножини, зміна порядку обчислень, дихотомічний пошук розв'язання задачі, розбиття задачі на підзадачі та інші. Всі ці способи використовуються при адаптації алгоритмів розв'язання задач на комп'ютерах. У процесі мінімізації часової складності алгоритму звичайно використовують одночасно декілька способів.

Формальні алгоритмічні системи. Основним призначенням формальних алгоритмічних систем ФАС є дослідження проблеми розв'язності. Для цієї проблеми вимога елементарності кроку є необхідною. Оскільки ця вимога не може бути математично точно сформульована, вона інтерпретується як умова загальної зрозумілості.

Прикладами математичних моделей ФАС є машина Тьюрінга, машина з безмежними регістрами, нормальні алгоритми Маркова, рекурсивні функції. Машина Тьюрінга розглядається з позицій метричної теорії, що дозволяє розширити перелік характеристик складності алгоритмів порівняно з неформальними алгоритмами. Крім часової складності, вона використовує також місткісну і програмну. Аналізуються такі способи мінімізації часової складності машини Тьюрінга:

- зміна розташування початкових даних на стрічці;
- вибір місця розташування проміжних результатів;
- вибір стратегії руху головки;
- вибір початкового положення головки;
- збільшення символів зовнішнього алфавіту;
- застосування паралелізму (багатострічкова машина Тьюрінга).

Наведені способи мінімізації часової складності, крім останнього, не мають практичного значення для комп'ютерної реалізації. Машина Тьюрінга є ідеалізованою моделлю алгоритму. Основним пунктом її ідеалізації, як і усіх інших математичних ФАС, є неврахування апаратних витрат, необхідних для реалізації алгоритму. Ця особливість математичних ФАС не дозволяє повною мірою використовувати досягнення теорії ФАС у проектуванні апаратно-програмних засобів. А у деяких випадках цей недолік приводить до практично неприйнятних висновків. Прикладом тому є теорема про лінійне прискорення. Для програмно-апаратних моделей алгоритму ця теорема не коректна.

Програмно-апаратна модель алгоритму (SH-модель). Відмінність комп'ютерних засобів від неформальних і формальних (абстрактних) моделей алгоритмів полягає у врахуванні апаратної складової реалізації алгоритму. Безпосереднє перероблення програмно-апаратної моделі алгоритму частково або повністю виконується апаратними засобами.

Відзначимо, що всі згадані раніше моделі алгоритму містять ті чи інші апаратні засоби або припускають їх наявність. У машині Тьюрінга під час звернення до програми припускається наявність дешифраторів при виборі тих чи інших символів зовнішнього та внутріш-

нього алфавітів. Але під час побудови ефективних алгоритмів апаратні затрати, пов'язані з дешифраторами, не враховуються. Нормальні алгоритми Маркова також містять підмножину елементарних розпізнавачів і перетворювачів, але в систему характеристик апаратні затрати не входять.

Використання апаратних засобів в програмно-апаратній моделі алгоритму дозволяє в процесі синтезу алгоритму використовувати п'ять взаємозалежних характеристик складності – апаратну, часову, програмну, структурну та місткісну.

Решта пунктів розділу розкривають:

- способи синтезу ефективних програмно-апаратних моделей алгоритму за окремими характеристиками складності;
- аналіз взаємозалежності характеристик складності;
- способи синтезу ефективних програмно-апаратних моделей алгоритму за оптимізованими характеристиками складності.

У зв'язку з великим значенням сучасних задач криптографії і оброблення зображень зв'язані з ними розділи теорії алгоритмів представлені на кафедрі ЕОМ НУ “Львівська політехніка” окремими курсами. Алгоритми розв'язання задач на графах віднесені до курсу “Основи автоматизації проектування”, в розділ “Конструкторське проектування”.

Висновки. Отже, програма-рекомендація пропонує для знайомства та засвоєння потужну множину неформальних алгоритмів з різних напрямків індустрії задач. Важливим є те, що навчання за цією програмою забезпечено значною кількістю літературних джерел. Наприклад, практично всі алгоритми, що є у програмі, описані в підручнику Кормена Т., Лейзерсона Ч., Ривеста Р. “Алгоритмы: построение и анализ”.

Наша програма має переваги у напрямках розширення і деталізації способів конструювання ефективних неформальних алгоритмів, у використанні програмно-апаратних моделей алгоритму, використанні п'яти характеристик складності і способів їх оптимізації.

Додаток 1. Попередній варіант програми-рекомендації для університетів США з дисципліни

“Алгоритми і складність”.

1. Базис аналізу алгоритмів (ядро – 4 год)

Асимптотичний аналіз верхньої та середньої границь складності. Відмінність між складностями в кращому, посередньому і гіршому випадках. Велике “О” і мале “о”. Класи складності (стандарт). Час і місткість. Використання рекурентних відносин в аналізі рекурсивних алгоритмів.

2. Алгоритмічні стратегії (ядро – 6 год)

Алгоритми “грубої сили”. “Жадні” алгоритми. “Поділяй і пануй”. Пошук з поверненням. Метод гілок і границь. Евристика. Машинні і вербальні алгоритми. Алгоритми числової апроксимації.

3. Фундаментальні комп'ютерні алгоритми (ядро – 12 год)

Прості числові алгоритми. Алгоритми з послідовним і бінарним пошуком. Бінарні пошукові дерева. Алгоритми сортування з вибіркою і вставкою [$O(n^2)$]. $O(N \log N)$ алгоритми сортування (швидке, купчасте, зливанням). Хеш-таблиці, що містять колізії - унікаючі стратегії. Представлення графів. Найкоротші алгоритми (алгоритми Дійкстри і Флойда). Перехідне закриття (алгоритм Флойда). Мінімально зв'язані дерева (алгоритм Пріма і Краскела). Топологічне сортування.

4. Розподілені алгоритми (ядро – 3 год)

Консенсус і вибір. Виявлення закінчення. Відмовостійкість. Стабілізація

5. Основи комп'ютерної теорії (ядро – 6 год)

Машини з кінцевим станом. Граматики вільно-контекстні. Необчислювальні функції. Проблема зупинок. Підтекст необчислювальності.

6. P- і NP- класи складності

Проблеми розв'язності і нерозв'язності. Визначення класів P і NP. NP-повнота (теорема Кука). Стандартні NP-повні проблеми. Технічне стиснення.

7. Теорія автоматів

Детермінований кінцевий автомат (DFA_s). Недетермінований кінцевий автомат (NFA_s). Еквівалентність DFA_s і NFA_s . Прості залежності. Лема закачування для простих залежностей. Магазинний автомат (PDA_s). Відношення PDA_s до контекстно-вільних грамастик. Машини Тьюрінга. Недетерміновані машини Тьюрінга. Набори (команд) і мови. Ієрархія Хомського.

8. “Просунутий” аналіз алгоритмів

Алгоритми в реальному часі і незалежні від часу. Амортизаційний аналіз. Ймовірнісні алгоритми. Динамічне програмування. Комбінаторна оптимізація.

9. Криптографічні алгоритми

Історичний огляд криптографії. Проблема власного ключа і ключа криптографії, що змінюється. Цифрова сигнатура (підпис). Протоколи (секретності) захисту. Використання (доведення з нульовим знанням, ідентифікація та інш.).

10. Геометричні алгоритми

Лінії сегментів (зображень): властивості, перетинання. Алгоритм виявлення опуклих поверхонь.

Додаток 2. Програма Львівської політехніки**1. Вербальне тлумачення алгоритму**

Історія розвитку поняття алгоритму. Уточнення інтуїтивного тлумачення алгоритму. Властивості, параметри, характеристики алгоритму. Структура алгоритму. Класифікація алгоритмів.

Визначення задачі. Пряма і обернена задачі. Структура задачі. Спільні та відмінні параметри і характеристики задачі та алгоритму.

Складність розв'язання задач. Асимптотична часова складність. Експоненціальна і поліноміальна складності. P- і NP- класи складності задач.

2. Неформальні алгоритми

Визначення. Моделі неформальних алгоритмів.

Алгоритм суцільного перебору можливих рішень – вихідний для конструювання ефективних алгоритмів.

Розбиття задач на підзадачі – розподіл сформульованого намагання, великі задачі: проектування комп'ютерних засобів, прогнозування погоди.

Параметри алгоритму як об'єкти мінімізації часової складності:

- зміна правила початку обчислень;
- зміна правила закінчення обчислень;
- розбиття масиву вхідних даних на підмасиви: дихотомія, балансування, бінарний пошук;

- розбиття множини можливих розв'язань задачі на підмножини: метод гілок та границь;
 - перетворення правила безпосереднього перероблення: еквівалентні перетворення, апроксимація, розбиття правила безпосереднього перероблення на підправила;
 - точні обчислення: модулярна арифметика, китайська теорема, алгоритми згорток простих чисел Мерсена, Ферма.
- Алгоритми сортування, характеристики складності.
Графові моделі алгоритмів сортування, часова та програмна складності алгоритмів.

3. Формальні алгоритмічні системи

Моделі ФАС. Розв'язувані задачі. Машини Тьюрінга: одно- та багатострічкові, детерміновані, недетерміновані. Визначення, властивості, параметри, характеристики складності, графічний образ машин Тьюрінга.

Нормальні алгоритми Маркова. Види підстановок. Принцип нормалізації. Характеристики складності.

Рекурсивні функції. Підстановка, примітивна рекурсія, мю-операція. Тезис Чьорча. Основна гіпотеза теорії алгоритмів. Теореми Гьоделя.

4. Програмно-апаратна модель алгоритму (SH-модель)

Недоліки математичних формальних алгоритмічних систем (ФАС). Колізії машиноорієнтованих ФАС і реальних комп'ютерних засобів.

Визначення, властивості, параметри SH-моделі. Технічні характеристики складності SH-моделі – часова та апаратна. Інформаційні характеристики складності SH-моделі – програмна та структурна. Місткісна складність. Визначення надлишковості часових діаграм та матриць інцидентів. Властивість ієрархічності.

Узагальнена складність комп'ютерних засобів. Взаємозалежність характеристик складності на різних ієрархічних рівнях комп'ютерних засобів: часової та апаратної, часової та програмної, часової та структурної, апаратної та програмної, апаратної та структурної, програмної та структурної. Взаємозалежність місткісної складності з іншими характеристиками.

Оптимізація характеристик складності в процесі синтезу апаратно-програмних комп'ютерних засобів. Вартісна вага характеристик складності. Способи мінімізації часової та програмної складностей.

1. Марков А.А. Теория алгорифмов. – М., – Л., 1954. – 375 с. 2. Математическая энциклопедия. Гл. ред. И.М. Виноградов. – М., 1977. 3. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов: Пер. с англ. – М., 1979. – 536 с. 4. Успенский В.А., Семенов А.Л. Теория алгоритмов: основные открытия и приложения. – М., 1987. – 288 с. 5. Трахтенброт Б.А. Алгоритмы и вычислительные автоматы. – М., 1974. – 200 с. 6. Мальцев А.И. Алгоритмы и рекурсивные функции. – М., 1986. – 368 с. 7. Катленд Н. Вычислимость. Введение в теорию рекурсивных функций.: Пер. с англ. – М.; 1983. – 256 с. 8. Вишенчук И.М., Черкасский Н.В. Алгоритмические операционные устройства и супер ЭВМ. – К.; 1990. – 197 с. 9. Ахо Альфред И., Хопкрофт Джон Э., Ульман Джеффри Д. Структура данных и алгоритмы: Пер. с англ., – М., 2000. – 384 с. 10. Стивенс Р. Visual Basic. Готовые алгоритмы: Пер. с англ. – М., 2000. – 384 с. 11. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ: Пер. с англ. – М., 2000. – 995 с. 12. Черкасский М. Складність апаратно-програмних комп'ютерних засобів // Сучасні проблеми в комп'ютерних науках. Зб. наук пр. – Львів, 2000. – С. 58-67. 13. Черкасський М.В. SH-модель алгоритму // Вісн. НУ “Львівська політехніка”, № 433. – 2001.