Міністерство освіти і науки України
Тернопільський державний технічний університет імені Івана Пулюя

Кафедра біотехнічних систем

Теорія алгоритмів

Конспект лекцій

Для студентів, що навчаються за спрямуванням вищої освіти 6.0804 — комп'ютерні науки
Спеціальність 7.080401 — інформаційні управляючі системи та технології

Тернопіль-2000

УДК 519.68

Яворський Б.І. Теорія алгоритмів/Конспект лекцій.— Тернопіль: ТДТУ імені Івана Пулюя, 2000.— 36 с.

У навчальному посібнику висвітлюються змістовні основи теорії алгоритмів для студентів технічних університетів, що навчаються за спрямуванням вищої освіти 6.0804 — комп’ютерні науки, спеціальності 7.080401 — інформаційні управляючі системи та технології.

Іл. 8, Бібліогр. 12 назв. + 6.

Рецензенти:

Грицик В.В : членкореспондент НАН України, д.т.н., проф., директор Державного науково-дослідного інституту інформаційної інфраструктури, м. Львів

Стухляк П.Д. д.т.н., проф., зав.кафедрою інтегровних комп′ютерних технологій, ТДТУ імені Івана Пулюя

Відповідальний за випуск:

Яворський Б.І зав. кафедрою біотехнічних систем ТДТУ імені Івана Пулюя

"ЗАТВЕРДЖЕНО"

на засіданні кафедри
біотехнічних систем
протокол № 9, від 22.11.2000 р.

ЗМІСТ

ВСТУП

Теорія алгоритмів (ТА) виникла з внутрішньої потреби математики. Математична логіка, метаматематика, алгебра, геометрія і аналіз залишаються і сьогодні однією з основних областей застосування теорії алгоритмів.

Інша область її застосування появилася в 40-х роках XX-го століття — при створенні ефективних (перш за все швидкодіючих) електронних обчислювальних машин (ЕОМ). Поява ЕОМ особливо сприяла розвиткові тих розділів ТА, що мали яскраво виражену прикладну спрямованість. Сюди насамперед належали алгоритмічні системи і алгоритмічні мови (основа теорії програмування) універсальних ЕОМ, цифрові автомати.

Саме поняття алгоритму тісно пов’язане з лінгвістикою, економікою, фізіологією мозку і психологією, філософією і природознавством тощо. Прикладом може послужити розумова чи практична діяльність людини у будь-якій сфері.

Підвищення ефективності технологій виробництва, транспорту, зв’язку, енергетики, медицини тощо, а також відбору, зберігання, поширення, створення відповідних знань потребували удосконалення та розвитку інформаційних (вимірювальних) та управляючих (керуючих) алгоритмів. Це успішно здійснюється на базі математичних моделей величин, сигналів, фізичних полів. Виникла теорія таких алгоритмів. Знайомство з її змістовними основами і є метою даного курсу лекцій.

Отже, для математика ТА — означення поняття алгоритму, алгоритмічні системи (рекурсивні функції, машини Тюрінга, нормальні алгоритми Маркова та ін.), формальні перетворення і оцінки алгоритмів, теорія автоматів і універсальні ЕОМ, формальна побудова і аналіз мов програмування тощо. Це відображено у відповідній літературі (деякі з книг наведено у списку літератури; книга Д. Кнут "Математические методы анализа алгоритмов" є базовою), напрямах наукових досліджень університетів (див. інтернет-сайти за пошуком з ключевим словом algorithm) та впровадженні їх результатів у практику (різноманіття "універсальних" процесорів, контролерів, конверторів тощо — наприклад, фірми Intel, Motorola, Analog Device, Texas Instrument).

Проте, проблеми життєдіяльності з кожним роком висувають задачі, розв’язання яких універсальними обчислювачами (процесорами, контролерами) неефективне. Основною проблемою постає не традиційні вибір чи створення мови програмування, написання програм, а "наскрізне" (від математичного моделювання прикладної проблеми до розрахунку геометрії та топологій інтегральної схеми спеціалізованого пристрою) проектування. Це означає, що інженерові необхідно вміти також й інше — оцінювати складність, часові параметри алгоритмів, будувати еквівалентні алгоритми, вміти робити їх тотожні перетворення задля пошуку кращого за певним критерієм алгоритму при заданих обмеженнях на технічні ресурси для його реалізації. Важливим є уміння вибирати чи будувати критерії оптимальності, тип перетворень та розуміння інтерпретації поняття існування алгоритму на базі задач оптимізації, розв′язаних в різних галузях життєдіяльності.

У даному посібнику теорія алгоритмів розглядається на концептуальних засадах теорії та засобів інформаційних та управляючих технічних чи людино-технічних систем, в рамках освітньо-професійної програми спеціальності "Інформаційні управляючі системи та технології".

Метою дисципліни є знайомство з засадничими поняттями математичного апарату, потрібного для постановки та розв’язування задач аналізу і оптимізації (за критеріями, побудованими за відповідними характеристиками інформаційних та управляючих систем певної області) алгоритмів. За основу ТА взято математично коректну та практично конструктивну енергетичну теорію сигналів та систем (ЕТСС). Потрібно тут зауважити, що у вже згаданій книзі проф. Д. Кнута алгоритм потрактовано поняттям оператора, яке застосовується й в ЕТСС. Але традиційна для алгоритмів теоретична база (комбінаторика, рекурентні співвідношення, асимптотичний аналіз) не дає можливості побачити це (а ще більше — використати) студентові та інженерові.

Під час вивчення цієї дисципліни студент повинен отримати знання про:

— математичні моделі і представлення інформаційних сигналів та систем керування;
— алгоритми роботи та побудови представлень інформаційних та керуючих систем;
— представлення алгоритму;
— стійкість алгоритмів, чутливість до малих змін їх параметрів;
— складність алгоритму;
— вплив на алгоритмічну розв’язність задач технічних засобів реалізації алгоритмів.

Після вивчення дисципліни студент повинен уміти:

— будувати критерії оптимальності алгоритмів;
— оцінювати складність алгоритму;
— виконувати тотожні перетворення алгоритмів з метою побудови конвеєрних, паралельних, систолічних та інших алгоритмів.

У студента виробляться також навички подання алгоритмів вербально (словами звичайної мови з використанням іі граматики), формулами, блок-схемою, графами, машинними мовами — Асемблера, високого рівня тощо, створення системи команд спеціалізованого процесора для реалізації ефективних алгоритмів та пристроїв інформування та управління.

Теоретичні положення курсу “Теорія алгоритмів” є основою для успішного засвоєння цілої низки спеціальних дисциплін, які вивчатимуться у подальшому.

ЛЕКЦІЯ 1

Тема: Основні поняття теорії алгоритмів

1. Означення алгоритму

Поняття алгоритму є одним з основних понять сучасної математики. Ще на початкових етапах її розвитку (Стародавній Єгипет, Вавилон, Греція) у математиці стали формуватися різноманітні обчислювальні процеси. За їх допомогою при розв'язуванні задач шукані величини обчислювалися послідовно з висхідних величин за певними правилами та інструкціями. З часом такі правила (інструкції) одержали назву алгоритмів.

Термін *алгоритм* походить з імені середньовічного вченого-математика Аль-Хорезмі, котрий ще в IX ст. (825 р.) дав правила виконання чотирьох операцій арифметичних дій в десятковій системі числення. Перелік виконуваних арифметичних дій був названий *ал-горизм*.

З 1747 р. замість слова *алгоризм* стали вживати *алгорисмус*, смисл якого полягав в комбінуванні чотирьох операцій арифметичного обчислення — додавання, віднімання, множення та ділення. Через століття для створеного Бебіджем комп'ютера, Адою Лавлейс було написано першу програму, що є першим технічним втіленням алгоритму.

До 1950 р. *алгорисмус* став *алгорифмом*. Смисл алгорифму найчастіше пов'язувався з алгорифмами Евкліда — процесами знаходження найменшої спільної міри (двох відрізків, найменшого спільного дільника двох многочленів тощо).

Аж до 30-х років XX століття поняття алгоритму мало методологічне значення. Його означували евристично.

Означення 1. Скінченна сукупність коректно сформульованих правил розв'язування задачі з класу (множини) задач називається алгоритмом.

Таке означення алгоритму не є строгим, математичним, оскільки у ньому не означено понять клас задач, правило (їх розв'язування). Протягом тривалого часу математики задовольнялися цим означенням, теорії алгоритмів не існувало. Не було серйозних випадків, коли математики розійшлися б в тлумаченнях стосовно того, чи є алгоритмом певний конкретно заданий процес.

Становище суттєво змінилося, коли технічні засоби математичних обчислень стали настільки потужними та зручними у практичному користуванні, що розв'язок задачі в аналітичному вигляді (у вигляді формули) виявився не потрібним. Уможливилося розв'язування задач, з невідомим розвя'зком, отримання якого в аналітичному вигляді було важко доступним. Почали бурхливо розвиватися обчислювальні (чисельні) методи розв'язування задач, і на перший план висунулася проблема доведення існування відповідних алгоритмів. Виявилося, що довести наявність чи відсутність алгоритму не все одно. Перше можна зробити шляхом фактичного опису хоч одного методу розв'язування. В цьому випадку достатньо Означення 1 поняття алгоритму, щоб переконатися, що описане є алгоритмом. Довести відсутність існування алгоритму таким шляхом неможливо. Для цього потрібно коректніше означити, що таке алгоритм.

В 20-х роках XX-го століття проблема означення алгоритму стала однією з основних, фундаментальних математичних проблем. Її вирішення було одержано у двох формах всередині 30-х років 20-го століття у роботах таких відомих математиків як Гільберт, Гедель, Черч, Кліні, Пост, Т'юрінґ на базі означень понять класів:

a) **арифметичних функцій** (які назвали **рекурсивними)**;
b) **процесу**.

Пізніше в роботах Маркова і Калужніна з'явилося інше трактування алгоритму — як особливої **відповідності** між **словами в абстрактному алфавіті** (з означенням понять алфавіт, слово, відповідність).

Отже, для коректного математичного означення алгоритму потрібно в свою чергу ввести, означити першорядні, математично строгі поняття клас функцій, рекурсивна функція, процес, клас процесів, абстрактний алфавіт тощо. З технічної, інженерної, практичної сторони цього не має потреби робити.

Означення 2. Скінченна сукупність операцій, якій притаманні властивості

a) дискретності — виконання чергової операції починається у певний момент часу, моменти часу не обов'язково еквідистантні;
b) детермінованості — перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції;
c) впорядкованості — встановлена черговість операції (інакше — відношення порядку на їх множині);
d) масовості — алгоритм розв'язує задачу, що належить до класу задач;
e) елементарності — завжди можна вказати елементарну (виконувану апаратно, не програмовану) операцію —

називатимемо алгоритмом.

Таким чином, алгоритм виникає при вирішенні проблеми (інформування, управління). Для цього будується математична модель проблеми і на її базі ставиться задача(і) та будуються методи її розв'язування. Вважатимемо, що алгоритм задано, математичні проблеми його існування, коректності постановки задачі, яку за його допомогою розв'язують, вирішено. А під ТА розумітимемо систему фактів про властивості відповідних математичних об'єктів, якими подають алгоритм, у тому числі про взаємозв'язки між ними (їх алгебру, структури; для детальнішого знайомства з цими поняттями потрібно звернутися до відповідних посібників та підручників, зокрема, конспекту лекцій з курсу дискретної математики, основ алгебри тощо). Особливо про такі взаємозв'язки, які підпадають означенню представлення. При цьому не забуватимемо про математичну коректність, зокрема, вербального, графічного, аналітичного та інших способів подання (визначення) алгоритму.

2. Подання алгоритмів

В інформаційних та керуючих технологіях для інформування та вироблення керівного впливу використовуються *вимірювання* — спеціальні процедури з вимірюваним об’єктом та мірою (однорідною з об’єктом зразковою величиною). Здавна відомий алгоритм Евкліда знаходження спільної міри двох взаємно несумірних відрізків (найбільшого спільного дільника двох чисел). Розглянемо технічні та інформаційні ресурси (*характеристики процесора* та *математичну модель розв’язуваної задачі*) для алгоритму Евкліда, його алгебричну структуру. Її знання уможливлює тотожні перетвори алгоритму.

2.1. Засоби подання алгоритму

Для розуміння властивостей вербального подання потрібно ознайомитися з основами лінгвістики. Наприклад: Соссюр, Фердінан де. Курс загальної лінгвістики. — Київ: Основи, 1998. — 324 с.

2.1.1. Математична модель. Задача знаходження найбільшого спільного дільника постала при вирішенні проблеми вимірювань — знаходження масштабу міри. Відомо, що вимірювання є певною процедурою, алгоритмом над вимірюваною величиною та мірою. При порівнянні об’єктів за значеннями їм властивої величини потрібно спочатку вирішити проблему вибору міри та знайти її значення. Для числових величин цю проблему вирішено постановкою та розв’язуванням задачі найбільшого спільного дільника.

Розглянемо цілі числа (натуральний ряд чисел). Всяке ціле, що ділить без остачі $a, b, \ldots, l$, називається їх спільним дільником. З *теорії подільності* відома теорема про ділення з остачею:

Теорема 1. Всяке ціле a представляється єдиним чином за допомогою цілого b рівністю $a = bq_0 + r_0; \quad 0 \le r_0 < b.$

Якщо взяти числа b та r_0, то отримаємо представлення $b = r_0 q_1 + r_1$. Далі, розглядаючи послідовно подібним чином числа $r_0, r_1; \ldots, r_{n-1}, r_n; r_n, r_{n+1}$, легко встановити, що коли $r_{n+1}=0$, то r_n буде найбільшим спільним дільником чисел a, b [див. Виноградов И.М. Основы теории чисел.— М.: Наука, 1981.— 176 с.].

2.1.2. Ресурси та система команд процесора. Потрібні ресурси процесора та система команд визначаються моделлю задачі та методом її розв’язання. Це диктує розроблення спеціалізованого процесора, "під задачу". Проте часто вибирається процесор "близький" до потрібного —"під клас задач" у найкращому випадку. Отже, нехай маємо процесор з (а) —ресурсами:

i. арифметичний пристрій з регістрами для результату q_0 та x_0 і регістрами x_1, x_2 для операндів;

ii. пристрій вводу;
iii. пристрій виводу,

та (б) — системою команд (множиною елементарних операцій арифметичного пристрою процесора):

i. перемножити операнди;
ii. ділити з відкиданням дробової частини результату (функція *ціла частина від числа*) та присвоїти знак мінус;
iii. присвоїти регістрові вміст іншого регістру;
iv. перейти на (відповідний) крок (конкретний зміст дивись в наведеному алгоритмі);
v. ввести число у регістр з пристрою вводу;
vi. вивести число на пристрій виводу;
vii. зупинити.

2.1.3. Лінгвістичні засоби подання алгоритму. Розглянемо лінгвістичні засоби на прикладі української мови (алфавіту, скінченної множини спеціально вибраних слів та граматики). Назвемо тому його вербальним (на словах) поданням. Отже, для вербального подання алгоритму використовують звичайну мову, її граматику — алфавіт, лексику, синтаксис тощо. При цьому використовується й семантика мови, виконуючи функцію коментаріїв. Але поліморфізм мови порушує однозначність слів — інформативних денотатів понять, якими означують операції та ін. складники алгоритму. Це знижує ефективність використання мови, зокрема, семантики при поданні алгоритму. Тотожні перетвори (алгебра) алгоритму забезпечуються граматикою мови, вибраної для його вербального подання, при врахуванні властивостей моделі задачі, методу її розв'язування, процесора тощо.

Засобами мови (вербальні) подаються усі операції та оператори (арифметичні, логічні) алгоритму.

2.1.4. Алгоритм (Евкліда). Користуючись обмеженою кількістю (множиною) слів (понять), які повно і несуперечливо означують поняття моделі задачі, систему команд процесора та означенням алгоритму побудуємо алгоритм Евкліда:

i. ВВЕСТИ більше число у регістр x_2;
ii. ВВЕСТИ менше число у регістр x_1;
iii. функція — ціла частина q_0 частки від ділення вмісту регістрів x_2 на x_1 зі знаком мінус;
iv. добуток x_0 вмісту регістрів x_1 та q_0;
v. сума x_0 вмісту регістру x_0 з вмістом регістру x_2;
vi. переслати вміст регістру x_1 у регістр x_2;
vii. переслати вміст регістру x_0 у регістр x_1;
viii. переслати нуль в регістр x_0;
ix. якщо x_1 не рівне нулеві — перейти на крок iii;
x. ВИВЕСТИ вміст регістру x_2.
xi. ЗУПИНИТИ.

ЛЕКЦІЯ 2

Тема: Формальне подання алгоритмів

1. Подання алгоритму блок-схемами

На практиці алгоритми подають за стандартом — у вигляді *блок-схеми*, рис. 1:

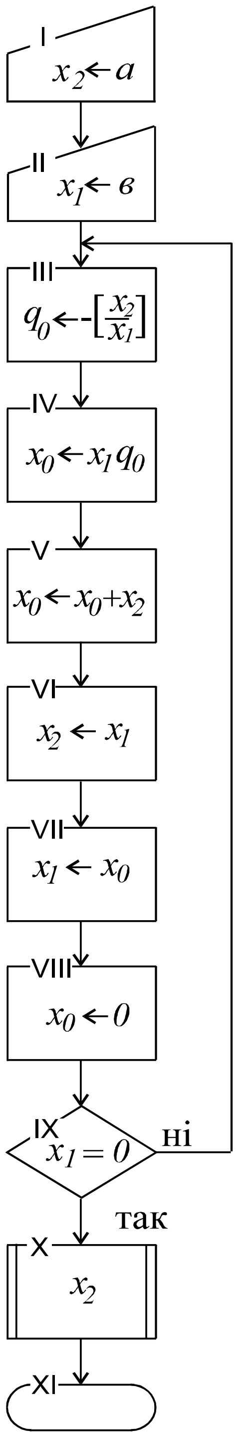


Рис.1. Блочна впорядкована схема алгоритму (номери блоків відповідають номерам кроків алгоритму підрозділу 2.1.4)

Стандартна блок-схема алгоритму в умовних позначеннях, виконана за певними правилами відображає порядок обчислень та їх зміст, логічні операції, операції вводу-виводу тощо. За умови збереження порядку обчислень, при фіксованих моделі задачі, методі її розв’язування та процесорі (системі команд) виконувати тотожні перетворення такої блок-схеми неможливо. Знімаючи ці обмеження, наприклад, при надмірних множинах команд процесора, моделей задачі на базі їх алгебр можлива побудова алгебри блок схем алгоритмів. Це уможливлює їх тотожні перетвори.

З теорії цифрових кіл (див., наприклад, Рабинер Л., Ґолд Б. Теория и применение цифровой обработки сигналов.—М.: Мир, 1978) відомі *функціональні* блок-схеми. У їх позначеннях подаються алгоритми обчислень (рис. 2):

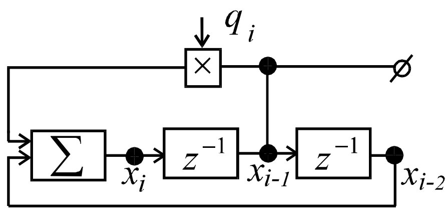


Рис. 2. Блочна функціональна схема алгоритму ($x_i, x_{i-1}, x_{i-2}, q_i$ — вміст регістрів $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{q}_0$ на i-му кроці обчислень; позначення всередині блоків: $\sum$ — суматор, $\times$ — помножувач, z^{-1} — обмін даними між регістрами на i-му кроці обчислень ("затримка" даних на час обчислень); стрілками показано функціональні "входи").

На функціональній блок-схемі алгоритму впорядкування обчислень позначують нумерацією її вузлів, яку виконують спеціально, використовуючи теорему про вигляд матриці впорядкованого цифрового кола. Логічних операцій ("галуження" обчислень) така блок-схема не відображає. Тому рис. 2 відображає лише частину алгоритму з рис. 1. На функціональній блок схемі (цифровому колі) існує алгебра лінійних тотожних перетворень. Крім того, існує ізоморфне його відображення та його алгебра, що надає великі можливості для розробки алгоритмів.

2. Аналітичне подання алгоритму

В аналітичному поданні алгоритму операції подають у вигляді формул, зі степенем дискретності, який відповідає елементарності операцій процесора. Ці формули повинні належати певній математичній структурі, щоби таким чином поданий алгоритм належав певній алгебрі (задля виконання тотожних його перетворів). Крім того, формули подають *впорядковано за часом їх виконання. Циклічність та галуження обчислень у цій послідовності подають за допомогою умовних, безумовних переходів, логічних операцій.*

Проаналізуймо властивості послідовності операцій алгоритму Евкліда:

i. $x_2 = a$;
ii. $x_1 = b$;
iii. $q_0 = -\left[\frac{x_2}{x_1}\right]$;
iv. $x_0 = x_1 q_0$;
v. $x_0 \leftarrow x_0 + x_2$;
vi. $x_2 = x_1$;
vii. $x_1 = x_0$;
viii. $x_0 = 0$;
ix. IF $x_1 \neq 0$ **THEN GO TO iii ;**
x. ВИВЕСТИ x_2;
xi. ЗУПИНИТИ.

Видно, що операції **i-iv** та **vi-viii** є операціями алгебри чисел. Інші операції (наприклад, **v** та **ix**) — ні. Це, а також впорядкування операцій за часом, не дозволяє означити алгебричну структуру, а, отже, й аналітичну модель алгоритму при заданій математичній моделі задачі, множині елементарних команд процесора.

Проблема існування аналітичної моделі алгоритму розв'язання задачі знаходження найбільшого дільника (двох цілих) чисел (необхідних та достатніх умов для цього) вимагає спеціального розгляду.

3. Подання алгоритму графом

Безпосередньо за функціональною блок-схемою будується граф цифрового кола, рис. 3:

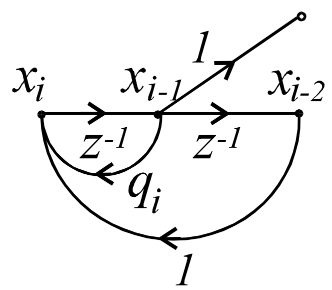


Рис. 3. Подання алгоритму у вигляді графу (цифрового кола).

Графи є об'єктом вивчення теорії графів (див., наприклад, Оре О. Теория графов.— М.: Наука, 1980, або Белов В.В., Воробьев Е.М., Шаталов В.Е. Теория графов.— М.: Высшая школа, 1976, чи конспект лекцій з дискретної математики). Формально граф алгоритму подається множинами вузлів v, направлених гілок h з "вагою" (деколи їх називають вершинами та ребрами) і предикатними відношеннями між ними. Для подання цих відношень використовуються матриці. Матрицею інциденцій $I = [V \times H]$ описують відношення між вузлами та гілками. Її елементами є 0 чи 1, залежно до відношення між відповідним вузлом та гілкою. Матрицею суміжності $R = [V \times V]$ описують відношення між вузлами. Її елементами r_{ij} є числа, якими подають число гілок між відповідними вузлами. Між цими матрицями існує зв'язок.

Використовуючи матриці інциденцій і суміжності, інші властивості графу обчислюють характеристики графу чи його частин — шляхів, дерев тощо. Ці характеристики мають свої інтерпретації та аналогії на характеристиках алгоритму (складність, швидкодія тощо). Наприклад, знаючи час виконання операцій, які відображені вузлами та гілками графу досліджуваного алгоритму, та використавши поняття шляху можна побудувати алгоритм обчислення часу роботи досліджуваного алгоритму; автоматичним підрахунком числа вузлів, гілок можна встановити складність алгоритму (надаючи певним операціям відповідну "вагу"). Матриці *R* та *I* використовують при цьому для подання алгоритму відповідною машинною мовою програмування.

Використовуючи функціональну інтерпретацію вузлів, гілок та їх характеристик за допомогою графів описують точність, робастність алгоритму (стійкість та чутливість до "збоїв" процесора та неточних даних чи параметрів алгоритму). Тут використовуються результати лінійної алгебри, теорії дискретних (цифрових) кіл. Зокрема, матриці *I* та *R* використовуються при розв'язуванні відповідних систем лінійних алгебричних рівнянь, що виникають. Складати ці рівняння немає потреби, бо вони автоматично задаються цими матрицями. Алгоритми їх розв'язування та програми також задаються відповідними програмними середовищами. Ці програмні середовища мають інтерфейс користувача (наприклад, "віконний"), за допомогою якого вказується потрібний вигляд аналізу (робастність, точність тощо).

На базі графів алгоритму проводяться також тотожні перетвори його структури. При цьому модифікуються матриці *I* та *R*. Ця модифікація є комбінаторною з використанням тотожних перетворів матриць та відношення порядку на множині вузлів. За таких перетворів отримують конвеєризацію операцій, їх розпаралелювання, виявлення систолік алгоритму тощо.

ЛЕКЦІЯ 3

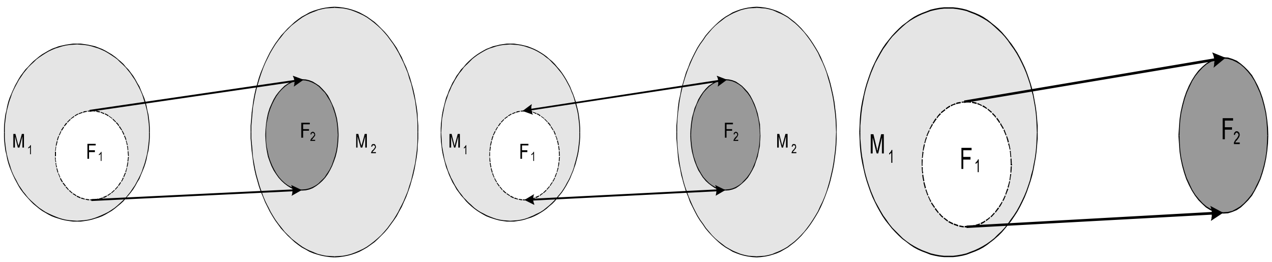
Тема: Представлення алгоритмів

1. Поняття про представлення

Зміст поняття "*представлення*" (representation) розглянемо на базі поняття *множини*. Розглядатимемо множини, яким властива певна *структура*. Структура задається властивостями операцій з елементами множини, результати яких є також елементами цієї ж множини (замкнутих операцій).

Між різними множинами розглядатимемо відображення. При відображенні структура може не змінитися. При цьому сподіватимемось на отримання практичних вигод від використання не самої множини, а її відображення.

Загальні випадки відображень множин поділяються на (рис. 4): **ін'єкції** (а) — відображення множини в підмножину іншої множини; **бієкції** (б) — взаємно однозначні відображення множин; **сюррекції** (в) — повне відображення однієї множини на іншу.



а) б) в)

Рис. 4. Відображення множин: (а) — ін'єкція; (б) — бієкція; (в) — сюррекція (M_1, M_2 — множини, F_1, F_2 — підмножини: $F_1 \in M_1,\ F_2 \in M_2,\ M_1 \cap M_2 = \varnothing$)

Між відображеннями існують відношення включення: сюррекція може бути ін'єктивною або бієктивною, бієкція — ін'єктивною, ін'єкція — несюрективна і небієктивна.

Коли вводиться поняття близькості між елементами множини (метрика), то відображення називають **морфізмами**. Їх позначають $\xrightarrow{F}$, де F — символ відповідного морфізму. Вони відповідно поділяються на **гомоморфізм**, **ізоморфізм**, **ендоморфізм**.

За допомогою спеціально (з практичних, інших міркувань) введеної функції-індикатора (індикаторної функції) множина розбивається на підмножини (**декомпозиція**); елемент множини **розкладається** (розвивається) через її примітивні елементи за допомогою операцій, заданих на цій множині.

Морфізми — конкретизований введенням метрики випадок відображень множин. Наприклад, для множини функцій зберігається норма, тобто

$$\int_T |\psi(t)|^2 d(t) = \sum_{k=\overline{1,K}} |\alpha_k|^2$$

де α_k — коефіцієнти розкладу функції $\Psi(t) = \sum_{k=\overline{1,K}} \alpha_k \varphi_k(t)$, $\varphi_k(t)$ — базисні функції (аналог примітивних елементів) функційного простору (множини функцій), для яких $\int\limits_{-\infty}^{\infty} \varphi_i(t)\varphi_j(t)dt = \begin{cases} 0, & i \neq j; \\ 1, & i = j. \end{cases}$.

Для не числових множин розглядаються спеціальні операції. Наприклад, для множин, елементами яких є вислови (**предикати**). Послідовність операцій називають виведенням (**продукцією**). Відображення тоді називають **реляціями** (відношеннями, позначаються символом ρ). Серед реляцій означуються **рефлексія**, **транзитивність**, **симетрія**. Якщо елементи множин пов'язані реляціями, які означуються кількома з цих реляцій, то розглядаються: **еквівалентність** (присутні всі три, позначається ~) та **порядок** (присутні перші дві, позначається $\square, \hbar$).

Вивід позначається у вигляді дробу, чисельник якого читається ЯКЩО, а знаменник — ТО. Відповідні відношення (реляції) позначаються:

1) рефлексія	$\dfrac{\alpha}{\alpha}$
2) транзитивність	$\dfrac{\alpha \xrightarrow{\rho} \beta \wedge \beta \xrightarrow{\rho} \gamma}{\alpha \xrightarrow{\rho} \gamma}$
3) симетрія	$\dfrac{\alpha \xrightarrow{\rho} \beta}{\beta \xrightarrow{\rho} \alpha}$

Тоді відношення еквівалентності та порядку означується наступним чином:

Еквівалентність	$\dfrac{(1) \wedge (2) \wedge (3)}{\sim}$
Порядок	$\dfrac{(1) \wedge (2)}{\square \vee \hbar}$

Відповідними до відношень еквівалентності та порядку на числових множинах є відношення рівності та більше-менше:

Рівність	$\dfrac{\alpha \in \mathrm{R} \wedge}{=}$
Більше-менше	$\dfrac{\alpha \in \mathrm{R} \wedge (\square \vee \hbar)}{< \vee >}$

Таким чином, для означення алгебри алгоритмів потрібні відношення та морфізми. Для представлення алгоритму потрібно означити відношення еквівалентності та ізоморфізм. Відношення порядку визначає можливість побудови (формування послідовності операцій) алгоритму та його вигляд і тотожні перетвори. Алгебра алгоритму є декомпозицією алгебр (а) — математичних структур, які означуються у моделі задачі та (б) — числення предикатів, яке означується за методами розв’язування задачі та вирішення заданої проблеми.

2. Теорія алгоритмів та її задачі

Теорія алгоритмів — система фактів про алгоритми, яка породжує нові факти. Розрізнятимемо у теорії алгоритмів подання алгоритму та його представлення.

Означення 3. Поданням алгоритму називається його опис.

Опис алгоритму можна здійснити засобами мови, блок-схем, формулами, графом тощо.

Означення 4. Представленням алгоритму називається ізоморфний перетвір його подання (якщо він існує).

Алгоритм містить як арифметичні, так і логічні операції. За їх допомогою перетворюються дані, відомості, а загалом — інформація.

Означення 5. Інформація — відомості про процес, стан (будь-якої природи) які сприймає людина безпосередньо чи через прилади.

Перетворення інформації подають через оператори. Оператору властиві його область визначення та область значень. Алгоритм у теорії означується як оператор. Важливим є поняття еквівалентності (рівності) алгоритмів.

Означення 6. Два алгоритми є рівними, коли співпадають їх оператори, області визначення та значень.

Означення 7. Два алгоритми є еквівалентними, якщо не співпадають їх оператори, але співпадають області визначення та значень.

Завдання теорії алгоритмів — вироблення засобів постановки задачі пошуку ефективних алгоритмів та методів її розв’язання: побудови критерію ефективності алгоритму, генерування його еквівалентних варіантів, подання обмежень ресурсів для його реалізації, побудови ефективних методів пошуку на множині алгоритмів його оптимального варіанту. При цьому важливим є існування у такій теорії ізоморфного представлення алгоритму.

ЛЕКЦІЯ 4

Тема: Генезис алгоритмів інформаційних та керуючих систем

1. Обчислювальні алгоритми

Життєдіяльність людини пов'язана з використанням та провадженням різноманітних явищ, процесів тощо. Кожному явищу властиві певні фізичні величини. Вони мають назви, параметри, характеристики. Часто їх можна визначати кількісно, міряти їх значення. Між різними величинами є взаємні зв'язки — фізичні закони (наприклад, Ньютона — в механічних явищах, Ома, Кірхгофа — в електричних, Фехнера — в фізіології). Математично закони подаються у вигляді формул. Формули отримуються й при розв'язуванні складених за цими законами рівнянь. Наприклад, значення в точці $t = nT_d$ розв'язку звичайного диференціального рівняння

$$\frac{d^2 y(t)}{dt^2} + c_1 \frac{dy(t)}{dt} + c_0 y(t) = x(t)$$

обчислюється за формулою ("згортки")

$$y(kT_d) = \sum_{i=1}^{n} a(iT_d) x[(k-i)T_d]$$

де $k \in \{0,1,2,3,....\}$; T_d — інтервал (період) відбору значень вхідного збурення x та відліків заникаючої функції a (ядра диференціального рівняння). При цьому потрібно знати n початкових значень x. На рис. 5 наведено граф-схему алгоритму цих обчислень. (Затримка на один такт обчислень аналітично позначається $z^{-1} = e^{-j2\pi f/f_d}$, де $f_d = 1/T_d$, $j = \sqrt{-1}$.) Пронумеровані точки позначають чарунки ("комірки") оперативної пам'яті обчислювача (значення коефіцієнтів a можуть також знаходитися в оперативній пам'яті).

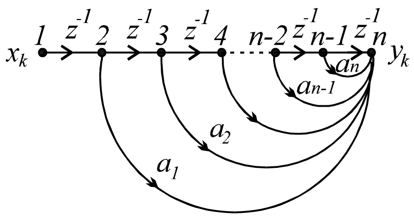


Рис. 5. Граф трансверсального алгоритму ("згортки", номери відліків винесено в індекс, позначення T_d при цьому не наводиться)

Класичні алгоритми, власне, пов'язані з обчисленнями значень якоїсь величини за значеннями інших величин. Вигляд зв'язку шуканої величини з даними величинами встановлюється заздалегідь (або емпірично, або аналізом встановлених раніше емпірично зв'язків). Досить скоро було зауважено, що від алгоритму обчислень на одному і тому ж засобі, однієї і тієї ж формули залежала їх точність, громіздкість, швидкість, чутливість результату до неточності даних тощо.

Означення 8. Алгоритм, у відповідності з яким розв’язування задачі зведене до виконання за правилами алгебри арифметичних дій називаються обчислювальними.

За допомогою обчислювального алгоритму шукається одне значення функції.

Обчислювальний алгоритм будується безпосередньо за формулою, з врахуванням правил чергування дій. Спосіб подання даних та результатів, виконання дій, визначення значень елементарних функцій тощо визначається за критерієм точність-швидкодія-затрати пам’яті, якщо це суттєво. Суттєвість побудови ефективного алгоритму та характеристики ефективності визначає його замовник. Оптимізацію алгоритму (обґрунтування критерію та його побудову, означення та визначення обмежень, побудову множини алгоритмів, пошук оптимального алгоритму) здійснює інженер-програміст на базі ТА — з врахуванням особливостей даних та формули (моделі задачі), ресурсів процесора, використовуючи модель подання алгоритму. Якщо оптимум не досягається, то змінюється постановка задачі (вибирається інша формула).

2. Поняття про математичну модель

Означення 9. Математичною моделлю називатимемо математичний об’єкт — формулу, функцію, рівняння тощо поставлені у відповідність явищу певної природи.

Математична модель повинна ефективно (повно, але у мінімальному об′ємі) відображати модельований об’єкт та відповідати задачам, які на її базі розв’язуються, тобто, має бути адекватною проблемі. Відповідність моделі до задачі означується достатніми та необхідними умовами існування її розв’язку, що уможливлює постановку задач. Ефективність моделі визначається також ефективністю (оптимальністю) алгоритму розв’язування задачі.

Постановка у відповідність явищу певної природи математичного об’єкту здійснюється шляхом моделювання. При цьому використовуються положення теорії розмірності, теорії подібності та фундаментальні фізичні чи інші принцип, закони.

Широко відомі: принципи — балансу ресурсів, найменшої дії, нерозривності (потоку), лінійності; закони — збереження (енергії, кількості руху, матерії тощо), сталого зв’язку між величинами. Вони, та положення і закони логіки Арістотеля, приводять до отримання згаданих математичних об’єктів. Їх отримують феноменологічно (на підставі прямого досвіду), евристично (на підставі правдоподібних, логічних міркувань — евристик) чи, навіть, інтуїтивно (здогадкою). Отримані вирази мають бути математично коректними, задовольняти математичним умовам існування (достатнім та необхідним). Крім того вони мають бути конструктивними — приводити до робастних, ефективних алгоритмів, числення.

Теорія подібності вивчає загальні властивості подібності явищ та їх моделей. Моделі розрізняють фізичні, аналогові та математичні. Співставлення, заміна явища, об'єкту відповідним явищем чи об'єктом (моделлю) цієї ж природи називається фізичним моделюванням. Коли модель іншої природи, ніж об'єкт моделювання, то вона називається аналоговою. Коли модель є математичним об'єктом, то моделювання називають математичним. Два явища подібні, якщо математичні моделі їх співпадають. Але цього недостатньо. Достатні умови подібності встановлюються коефіцієнтами подібності. Встановлено для явищ різної природи системи аналогій (електромеханічних, електро-теплових, механічно-теплових та ін.). Таким чином, ще один спосіб математичного моделювання — використання систем аналогій (об'єкт-модель) та відповідних математичних результатів у складнику-моделі системи.

Теорія розмірності вивчає категорію розмірності величин, властивих явищу. Множина цих величин неоднозначно (традиційно, законодавчо тощо) розбиваються на базові та похідні підмножини. Тоді встановлюються формули визначення похідних величин через базові з використанням формул розмірних залежностей. Це використовують при моделюванні.

Математичні моделі відіграють базову роль при створенні у коректних та конструктивних алгоритмів, технологій інформування та керування.

3. Числові алгоритми

<u>Означення 10</u>. Алгоритми, у відповідності з якими шукаються значення функції називатимемо числовими.

Числові алгоритми будуються за методами розв'язування задач математичної фізики (крайових задач — математичних моделей явищ). Ці задачі поставляють у вигляді рівнянь у частинних похідних (рівняння хвиль, дифузії, потоків тощо у різних середовищах, різної природи) при відповідних початкових та граничних умовах.

При послідовному обчисленні формули згортки отримаємо значення функції (яка відповідає заданому диференціальному рівнянню). При цьому виникає потреба встановлення відношення порядку на множині номерів тактів обчислень.

Для простого диференціального рівняння числовий алгоритм будують також за рекурсивною формулою

$$y_k = \sum_{i=1}^{2} b_i y_{k-i} + x_k$$

де k відповідає nT_d. Граф-схема алгоритму наведена на рис. 6.

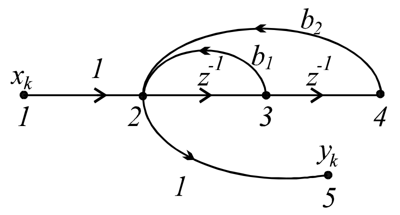


Рис. 6. Граф рекурсивного алгоритму (вузол 3 відповідає y_{k-1}, а вузол 4 — y_{k-2})

Аналітичне подання алгоритму повністю неможливе: налаштування процесора, процедури вводу-виводу, умовних та безумовних переходів вимагають подання у вербальному вигляді. Крім цього, оператори присвоєння значень відповідних даних (результатів обчислень, вмісту чарунок пам’яті тощо) регістрам процесора не відповідають означенню знаку "дорівнює".

001	**К=0; Y1=0; Y2=0; В1=1,8; В2=0.9;**	**% Налаштування процесора. К — % номер кроку обчислення, Y1, Y2 — % чарунки пам’яті, нулеві початкові % умови; B1, *B2* — коефіцієнти % різницевого рівняння**
002	***XK=x*** x_K**;**	**% Введення значення відліку вхідного % збурення (сигналу) з АЦП**
003	***YK=XK+Y1*B1+Y2*B2*** **;**	**% Обчислення відліку шуканої % функції**
004	***Y2=Y1*;**	**% Затримка на один такт обчислень**
005	***Y1=YK*;**	
006	**y_k=YK;**	**% Вивід відліку шуканої функції на % ЦАП**
007	**K=K+1**	**% Лічильник номеру кроку обчислень**
008	**GO TO 2**	**% Безумовний перехід на 2-й крок % алгоритму**

ЛЕКЦІЯ 5

Тема: Алгоритми інформаційних та керуючих систем

1. Алгоритми вимірювання

Для обчислень значень інформативних величин, для керування потрібні дані. Дані отримуються за допомогою спеціальних процедур (алгоритмів) над відібраними від явища значеннями властивих йому величин та їх мірами. Об'єктом виміру є властивість. Маса, колір, електричний опір, попит на товар, купівельна спроможність, розумова спроможність тощо. Властивості відносять до емпіричних явищ — людей, речовин, фізичних полів. Один об'єкт має кілька властивостей. Вимірюючи одну властивість нехтуємо іншими. Властивості перебувають у взаємозв'язках. Тому можливі непрямі вимірювання. З точки зору результату виміру різні об'єкти можна вважати еквівалентними, якщо він однаковий. Властивості об'єктів моделюють математичними об'єктами. У цьому сенсі вони наділяються математичною структурою.

Означення 11. Гомоморфне відображення властивостей виміру на математичний простір називають вимірювальною шкалою.

Теорія шкал базується на теоретико множинному апараті відношень. Множина величин S з емпірично встановленими відношеннями між ними R_i складає систему $E:\{S,R_i\}$. На практиці досить зручно користуватись числовою системою $N:\{M,P_i\}$, де M — множина чисел, P_i — відношення між числами. Тобто, потрібно емпірично встановлені пари величин пов'язати з парами чисел. У теорії шкалу подають трійкою $\langle E,N,\Psi\rangle$, де Ψ — гомоморфне відображення (в один бік) E на N.

Відомі у застосуваннях (розпізнавання образів, побудова емпіричних залежностей тощо) такі шкали — абсолютна, відношень, інтервалів, порядку, назв.

Абсолютна шкала є метричною, без перетворень. Наприклад, вимірювання характеристик мікросвіту (енергії кванту) за її макро- проявами (частотою випромінювань ν) здійснюють за співвідношенням $E=h\nu$, відомим у квантовій механіці (тут h — константа).

Шкала *відношень* (пропорційна) є метричною з перетвореннями (подібності — зсуву, масштабу тощо). Наприклад, знаючи коефіцієнт k для пружини можна міряти силу F за її розтягом x використавши співвідношення $F=kx$ (закон Гука).

Шкала *інтервалів* метрична частково, допускає перетворення вигляду $\varphi(x)=kx+b$. Наприклад, відомі перетворення температурних шкал (Фаренгейта — Цельсія, Реомюра — Цельсія тощо).

Шкалу *порядку* називають ординальною. У цій шкалі деякі об'єкти стають еквівалентними. Допустимими перетворами такої шкали є перетворення, які не змінюють порядку.

Шкала *назв* (номінальна) найбільш слабка. Тут числа служать умовними назвами об'єктів вимірювань. Допустимими перетвореннями такої шкали є пермутаційні (перестановки).

Між метричними шкалами існують взаємні перетворення та відношення. Найбільш сильною шкалою є абсолютна шкала (вона включає в себе всі інші шкали).

Означення 12. Шкала — алгоритм гомоморфного відображення емпіричної системи з відношенням у числову систему.

На теоретичному формальному рівні вимірювання трактується лінійним перетворенням, оператором. Це трактування формально означується як еквівалентність (процедури вимірювання, фільтрації). Фільтр математично означується оператором. За виглядом розрізняють диференціальні, інтегральні, інтегрально-диференціальні оператори. Диференціальним оператором є, наприклад, звичайне диференціальне (та його дискретний аналог — різницеве) рівняння. На рис. 7 подано алгоритм вимірювання.

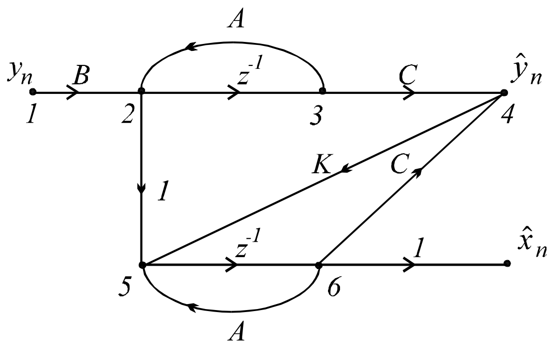


Рис. 7. Алгоритм Калмана-Бюсі оптимального вимірюваня значення випадкового процесу (*A, B, C* — задані матриці, елементи яких є параметрами вимірюваного об'єкту; y_n — вимірюваний стан об'єкту, вектор; x_n — оптимальна оцінка вектора стану; *K* — коефіцієнти Калмана, шукають в процесі обчислень за відповідним алгоритмом)

Для зашумлених даних (суміші даних та шуму) будують оптимальні алгоритми оцінювання значень вимірюваної величини. До таких алгоритмів належить, наприклад, оптимальний фільтр Колмогорова-Вінера. Його реалізують у вигляді трансверсального алгоритму (рис. 5), параметри a_i, $i=1, 2,3,\ldots n$ якого шукають розв'язуючи відповідну задачу, постановка якої передбачає відомий вигляд суміші (адитивна) даних і "шуму" та відомі їх статистики (кореляційні функції чи спектри). Іншим є оптимальний фільтр Калмана-Бюсі. Алгоритм його роботи має характер ітерацій при пошуку кожної з оцінок значень даних.

Задавши об`єкт формальним описом (математичною моделлю) у просторі його станів вектором *y,* (компоненти вектора — величини, властиві об'єкту), результати вимірювання їх значень позначивши вектором *x,* а вплив різних факторів на стан хворого і виміри їх значень — *u* та *w* відповідно, отримаємо

$$\begin{aligned} \frac{dy}{dt} &= Ay + Bu \\ x &= Cy + Dw \end{aligned},$$

де матриці відповідно визначають спосіб дії: *B, D* — впливів, *C, A* — вимірів та станів. Процес встановлення конкретного вигляду всіх цих позначень називають математичним моделюванням. Зокрема, їх певний вигляд відповідає постановці задачі фільтрації Калмана-Бюсі — оцінювання стану (об'єкту). Калманівська фільтрація є моделлю оцінювання (вимірювання) стану. Якщо процесор, виконує обчислення, необхідні для фільтру (розрахунок коефіцієнтів *K* розв`язуванням рівняння Рікаті тощо), то отримаємо *автоматичного оптимального спостерігача* (технічного засобу, що виконує одну з функцій людини — оцінювання стану керованого об'єкту). Для дискретних значень компонент векторів дістанемо модель у просторі станів

$$\begin{aligned} y_{n-1} &= Ay_n + Bu_n \\ x_n &= Cy_n + Dw_n \end{aligned},$$

де y_{n-1} — вектор з компонентами різниць першого порядку (дискретний аналог похідних). Для цієї моделі алгоритм пошуку компонент вектора подано на рис. 7 (де вплив шумів, похибок на вимір компонент вектора станів x_n відсутній).

2. Алгоритми керування

Виміряні дані використовують для керування. Поняття керування виникло давно. Наприклад, вже у стародавньому Китаї, Єгипті, древній Греції та старому Римі розуміли роль зворотного зв'язку та поінформованості при забезпеченні стабільності та потрібного розвитку різних процесів. Проте теорія керування розвинулася лише протягом XIX та XX століття, а практично широке застосування його розвинулося у другій половині XX століття.

Загалом керування розрізняють: (а) програмне (за наперед відомим алгоритмом з постійними параметрами); (б) автоматичне, коли враховуються вхідні дані; (в) адаптивне (коли параметри алгоритму змінюються залежно від зовнішніх умов, даних). Адаптивне та автоматичне керування здійснюють використовуючи зворотній зв'язок. Коли, враховуючи різні обмеження, характеристика алгоритму автоматичного чи програмного керування за встановленим критерієм найкраща, то керування називають оптимальним.

Поняття зворотного зв'язку є одним з фундаментальних у кібернетиці — науці про ефективне керування (розрізняють технічну кібернетику та інші: медичну, криміналістичну, спортивну й т.п. — див., наприклад, книгу С. Бир Мозг фирмы.—М.: Радио и свъязь,1993. 416 с.). Іншим фундаментальним поняттям цієї науки є інформація. Воно безпосередньо розвивалося в теорії комунікації (зв'язку). Ці поняття й розглядатимемо у контексті, прийнятому в цих технічних науках.

В широкому розумінні інформація — це якісь відомості, дані про конкретне явище. У техніці її, як правило, пов'язують зі змінною фызичною величиною, сигналом (та його математичною моделлю — функцією). Сигнал є фізичним носієм інформації. Таким чином, вимірюючи параметри та характеристики сигналу отримуємо інформацію якщо вони міняються, а їх інтерпретація відома.

Розглянемо елементарну систему керування (спеціальним чином з'єднані пристрої, рис. 8). Феноменологічно такі системи були зауважені у природі ще в древності. Задля забав їх виконували (імітували, повторювали) різними технічними засобами. Пізніше вони використовувалися у годинниках, парових машинах в регуляторах (стабілізаторах) швидкості обертання валу механізму.

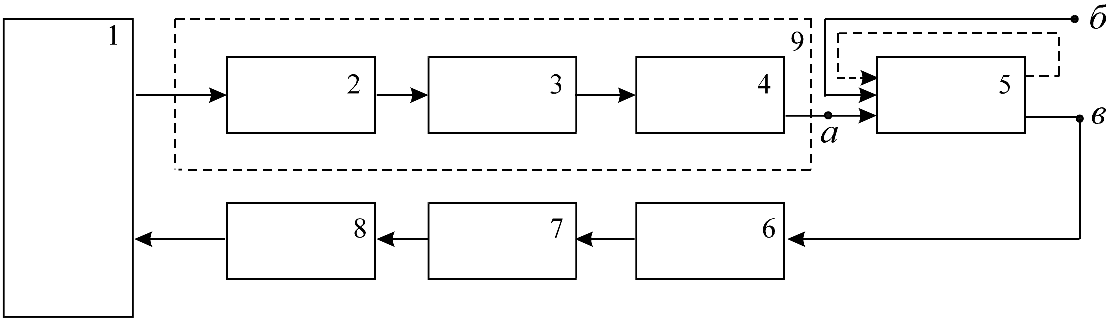


Рис. 8. Блок-схема системи керування.

Елементарна система керування складається з ланок: пристрою порівняння (5), регулятора (6, 7, 8), зворотного зв'язку (2, 3, 4). Сигнал $x(t)$ від керованого об'єкту (1) та відповідні характеристики ланок системи у лінійній теорії сигналів та систем подають як функції комплексної змінної $p=j\omega$, де $\omega=2\pi f$ (інакше – як лапласівські образи функцій часу, який є дійсною змінною), f — частота, *гц*. Нагадаймо, що для ланок системи цими характеристиками є тоді лапласівські образи їх вагових функцій (відгуків на дельта-збурення) — функції передачі регулятора $K_p(p) = K_6(p)\ K_7(p)\ K_8(p)$ та зворотнього зв'язку $K_{зв}(p) = K_2(p)\ K_3(p)\ K_4(p)$. Це приводить до зручнішого подання залежності між вхідним сигналом $X(p)$ (на вході блоку 2) та вихідним — на виході блоку 8, $Y(p)$:

$$Y(p)=(X(p)+Y(p)K_{зв}(p))K_p(p)\ ,$$

або у наочнішому для користування вигляді

$$Y(p)=X(p)\frac{K_p(p)}{1-K_{зв}(p)K_p(p)}\ .$$

Вирази $K_{р,зв}(p)$ є алгебричними поліномами вигляду $\sum_{i=1}^{N} k_i\, p^i$ (вони називаються функціями передачі). Від їх вигляду (порядку N, значень коефіцієнтів k_i) залежить стійкість системи.

Для дискретних систем (і в основному й для цифрових, коли дискретні значення сигналів кодуються двійковим чи іншим кодом) матимемо подібний вираз, але як функцію від $z = \exp(j2\pi\, fT_d)$, де T_d — період дискретизації неперервного сигналу (фіксовані інтервали часу для виконання операцій в алгоритмі)

$$Y(z) = X(z)\frac{K_p(z)}{1 - K_{зв}(z)\,K_p(z)}\ .$$

Цей вираз є розв'язком системи лінійних рівнянь, складених за рис. 8 з використанням законів дискретних кіл (мереж) чи властивостей відповідного йому графу. Таким чином, чисельник цього виразу є доповненням до елементу з індексами рівними номерам вузлів входу-виходу матриці системи рівнянь, а чисельник — детермінантом цієї системи.

Функції $K_{р,зв}(z)$ у загальному випадку дробово-раціональні, їх обчислення виконується за алгоритмами, подібними на алгоритм ланки управління. Вони за структурою подібні до алгоритмів вимірювання, див. рис. 5-7.

На множині номерів вузлів дискретного (цифрового) кола моментом часу обчислення задається відношення порядку. Множина вузлів розбивається на підмножини (за відношенням еквівалентності). В свою чергу, на цих підмножинах також задається часове відношення порядку. Виникають тим самим умови для паралелення та конвеєризації обчислень. В алгоритмах векторно-матричних обчислень спостерігається залежність кількості зайнятих в обчисленнях ресурсів від часу, яку за аналогією з хвилями в кровоносній системі організмів називають систолічністю. Структури, в яких з метою підвищення їх ефективності враховуються ці особливості, називають систолічними. Ці властивості використовують під час побудови оптимальних за швидкодією чи кількістю обладнання алгоритмів.

ЛЕКЦІЯ 6

Тема: Математичні проблеми теорії алгоритмів

1. Алгоритмічна розв’язність задачі

За означенням алгоритмові притаманна властивість масовості — він надається для розв’язування будь-якої задачі з класу задач породжених проблемою. Проблема може бути алгоритмічно не розв’язною. Це не означає, що немає хоч би одної задачі, для якої можна побудувати алгоритм її розв’язування і вирішити проблему.

Прикладом алгоритмічно розв’язної проблеми є доведення тотожності виразів в алгебрі чисел. Послідовним зведенням подібних членів, розкриванням дужок тотожність будь-яких виразів, заданих в алгебрі, доводиться за скінченну кількість кроків. Результатом алгоритмічної розв’язності проблеми (класу задач) є розроблення спеціальних мов програмування чи спеціалізованих процесорів аналітичних перетворень алгебричних виразів. При цьому виникає проблема оптимального розподілу функцій між апаратурою та програмами для аналітичних (тотожних) перетворень алгебричних виразів. Її вирішують для класів виразів, отриманих на базі властивостей конкретних інформаційних моделей.

Для дослідження алгоритмічної розв’язності проблем було введено поняття машини Тюрінга та рекурсивної функції. На базі цих понять було доведено (1936 рік, Черч) алгоритмічну нерозв’язність проблеми логічного виводу (існування алгоритму зведення будь-яких виразів в алгебрі логічних змінних). Відома теорема Геделя про неповноту логіки арифметики у якій є істинні твердження, які отримати неможливо логічними перетворами інших тверджень арифметики.

Традиційний метод доведення існування алгоритму — використання машини Тюрінга і "методу від супротивного", щоб показати існування рекурсивності (рекурсивної функції). Так доводиться алгоритмічна розв’язність звичайних диференціальних рівнянь. При цьому вважається, що складніший процесор можна побудувати на базі машини Тюрінга, що дозволяє його використати для доведення.

1.1. Математичні моделі обчислювальних апаратів

Для досліджень властивостей, встановлення закономірностей, характеристик алгоритмів при вирішені внутрішніх проблем ТА було введено гіпотетичні (умовні) обчислювальні машини. Це було зроблено незалежно одним від одного в 1936-1937 рр. Постом та Тюрінгом. Евристично вони побудували машини достатньо прості, але такі, що виконують всі відомі алгоритми. У подальшому ці машини не розрізнялися і їх стали називати машинами Тюрінга.

Машина Тюрінга складається з:

(a) пам'яті у вигляді нескінченного ланцюжка чарунок (комірок) на один символ;

(b) пристрою читання вмісту чарунки;

(c) пристрою управління з скінченним числом станів (стани машини).

Чарунки пам'яті пересуваються біля пристрою читання так, що у кожен певний фіксований момент часу він знаходиться проти певної чарунки.

Постом запропоновано алгоритмічну систему у якій дані у пам'яті подаються одним з символів алфавіту $A = \{0,1\}$. Система команд містить команди:

(a) записати в чарунку 1 і перейти на i -у команду;

(b) записати в чарунку 0 і перейти на i -у команду;

(c) перейти до чарунки зліва і виконати i -у команду;

(d) перейти до чарунки справа і виконати i -у команду;

(e) якщо в чарунці 1, то перейти до виконання j -ї команди, якщо в чарунці 0, то перейти до виконання i -ї команди;

(f) зупинка.

Алгоритм з командами машини Поста називають алгоритмом Поста. Доведено, що будь-яку рекурсивну функцію можна подати алгоритмом Поста.

У машині Тюрінга пристрій управління може знаходитися в одному з скінченних станів $Q = \{q_0, q_1, \ldots, q_m\}$, а чарунка містить один символ із скінченного алфавіту . $A = \{s_0, s_1, \ldots, s_m\}$.

1.2. Рекурсивні функції

Означення 13. Рекурсією називається спосіб завдання функції, при якому біжуче значення функції визначається через попередні для меншого аргументу.

Означення14. Числова функція, значення якої можна обчислити називається вичислимою.

Означення 15. Процедура обчислень називається ефективною, коли існує її алгоритм.

Означення 16. Сукупність числових функцій співпадаюча з сукупністю ефективно вичислимих функцій називається сукупністю рекурсивних функцій.

Теза Черча. Клас рекурсивних функцій тотожний класові скрізь означених вичислимих функцій. (Її довести неможливо).

Ця теза є базовою при доведенні алгоритмічної розв'язності проблем. Досить довести математичними засобами, що розв'язок задачі не можна подати рекурсивною функцією, щоби показати її алгоритмічну нерозв'язність.

Рекурсивне подання функції алгоритму подано у лекції 4 про генезис алгоритмів інформування та управління.

2. Алгоритмічна розв'язність задач інформування та оптимального керування

Проблему інформування на змістовному рівні сформулюємо як: а) однозначне, мінімальне за обсягом, зручне для користування відображення даних; б) однозначне, мінімальне за кількістю інформаційних елементів представлення (ізоморфізм) даних. За п. (а) інформування майже завжди виконується для людино-машинних систем (ергатичних). При цьому відображення як правило має вигляд графіка (у т. ч. тривимірного, кольорового, динамічного) на екрані дисплею. За п. (б) інформування виконується для автоматичних (машинних) систем, що визначає фізичний спосіб його подання (у вигляді кодів у пам'яті комп'ютера тощо). Можливі комбінації вказаних способів подання інформації.

Алгоритмічна розв'язність проблеми інформування означується умовами існування (достатніми й необхідними) вказаних інформаційних елементів (графіків, ізоморфних представлень тощо). Ці умови потрібно шукати у теорії відповідних математичних моделей. Проблема інформування розв'язується за допомогою розв'язування задач пошуку інформаційних елементів за вимірами величин. Задачі формулюються на базі згаданих математичних моделей.

Проблема оптимального керування на змістовному рівні формулюється як вироблення керівного впливу на змінні стани керованої системи, який з мінімальною середньоквадратичною похибкою приводить до бажаного стану системи.

Алгоритмічна розв'язність проблеми керування означується умовами існування (достатніми й необхідними) керівного впливу. Ці умови потрібно шукати у теорії математичних моделей керованої системи та збурень, яким вона піддається. Проблема оптимального керування розв'язується за допомогою розв'язування екстремальних (деколи їх називають — оптимізаційних) задач пошуку керівного впливу за вимірами значень станів керованої системи, її збурень, значень середньоквадратичних відхилень стану та технічних обмежень. Задачі формулюються на базі відповідних математичних моделей.

3. Тотожні перетворення алгоритмів

Перетворення алгоритмів базуються на алгебрі структури математичної моделі задачі та алгебрі структури алгоритму. Важливими для практики перетвореннями є тотожні перетворення алгоритму. Їх провадять з врахуванням властивостей тотожних перетворень обох згаданих алгебр.

Найпростішим є алгоритм у вигляді послідовності елементарних операцій (команд з системи команд процесора) без галужень. Тип операцій визначається математичною моделлю, методом розв′язування задачі та множиною операцій процесора.

Черговість виконання операцій визначається алгеброю, заданою на базі математичної моделі. Тому на множині операцій означено відношення: порядку та еквівалентності за індексами операцій (кроками алгоритму). Відношення порядку встановлює черговість операцій. Відношення еквівалентності надає можливість: (а) — одночасно виконувати кілька операцій (паралелити їх виконання); (б) — міняти черговість їх виконання. Це важливо, бо веде до: (а) — зменшення часу, та (б) — похибки обчислень. Міняти черговість виконання операцій (у тому числі й паралелити їх) можна також перетвореннями алгебри структури алгоритму, якщо це не суперечить відношенню порядку.

Алгебри структури алгоритму та моделі дозволяють укрупнювати чи розбивати операції. Це використовується під час оптимізації алгоритмів (з врахуванням вже сказаного). При цьому окрім циклічного групування операцій (каскадування) виникає відношення порядку на множині циклів (каскадів). При циклічних обчисленнях та змінах порядку чергування циклів досягається підвищення загальної точності обчислень.

Для циклічного виконаннях алгоритмів у процесорах використовуються різні типи адресації даних. При цьому можна отримати підвищення швидкодії за рахунок зменшення пересилок даних, виконуючи операції над їх індексами.

Ще одна можливість у перетвореннях алгоритму отримується при врахуванні особливостей даних — їх періодичності тощо. Тоді виникають алгоритми зі змінами кількості ресурсів, зайнятих в обчисленнях (систолічні).

Крім однопослідовних обчислень (каскадів, лінійних, або — конвеєрних), паралельних в рамках каскаду чи конвеєра, систолічних для розв′язування задач використовуються умовні переходи на інші обчислення, у тому числі — на початок проведених вже обчислень. Такі переходи (за умовою) називають зворотніми зв′язками, а за виглядом структури алгоритму — петлями.

Паралельні гілки обчислень можуть можуть бути конвеєрами, систолічними структурами. Зустрічаються інші часово-просторові складні комбінації структур в алгоритмі. У цьому випадку розглядають декомпозицію структури алгоритму на окремі підструктури.

4. Операторні системи алгоритмів

Математичні моделі вимірювання та управління породжують відповідні алгоритмічні системи. Набір засобів системи алгоритмів вважається незмінним — при поданні алгоритму, його виконанні тощо. У системі вводиться поняття виконання алгоритму (наперед встановленими, впорядкованими командами). Всі команди розпадаються на логічні та арифметичні. Арифметичні операції безпосередньо перетворюють інформацію, логічні — визначають напрям цих перетворень. В машині Тюрінга логічною операцією вважають рух стрічки вправо чи вліво залежно від стану машини і прочитаного символу. Проте, для алгоритмів керування та інформування такі логічні операції тривіальні і побудова на їх базі потрібних переходів, логічних операцій приводить до громіздкості подання відповідних алгоритмів. Ця проблема вирішується введенням спеціальних систем алгоритмів (Ван Хао, Ляпунова, Янова тощо — див. кн. А.П. Ершов Теоретическое программирование).

ЛЕКЦІЯ 7

Тема: Технічні параметри алгоритмів

1. Характеристики та параметри алгоритмів

Характеристика алгоритму загалам є функцією багатьох змінних з параметрами, що хх описують. До технічних параметрів віднесемо точність та чутливість (їх часом об′єднують однією назвою — робастність) алгоритму. Їх можна визначити (а) — експериментально, (б) — розрахунком. Швидкодію, габарити прцесора, споживану енергію, вартість (апаратні ресурси) оцінюють за фактом реалізації алгоритму та розрахунком, коли відомі засоби для його реалізації. Ці параметри називають техніко-економічними. Розрахунок параметрів алгоритмів має перевагу перед експериментальними випробовуваннями, оскільки останні завжди дорожчі. Проте він у кінцевому результаті не завжди (наприклад, під час розроблення нової техніки) точно відображає реальність. Тому традиційно розрахунок носить орієнтовний характер і у подальшому уточнюється натурними випробовуваннями. Це зменшує загальні затрати на розробку. Розрахунок точності, чутливості, швидкодії виконується за відомою моделлю алгоритму та характеристиками і параметрами відповідних операцій.

1.1. Т о ч н і с т ь . Точність алгоритму означується стандартним чином, на базі математичної моделі похибки і понять норми та метрики. У загальному випадку похибка має складові — моделювання, методичну, інструментальну, обчислювальну тощо. Тому потрібно узгоджувати їх моделі так, щоби загальну похибку складала їх сума (зважена лінійна чи квадратична). Похибку обчислень визначають за похибками відповідних операцій над дискретизованими, квантованими та обмеженими у діапазоні значень числами, при перетворах їх кодів тощо). Інструментальною складовою похибки нехтують. Модель задачі та метод її розв’язування визначають відповідні операції, їх порядок і тому впливають на значення похибки.

Означення 17. Похибкою називається різниця між точним та обчисленим значеннями величини.

За точні значення та результати беруться не цифровані (дискретизовані та кодовані) значення. Послідовності похибки обчислення цифрових значень величин розглядаються (моделюються) у залежності від контексту — як значення випадкової величини чи послідовності. Звідси, основні характеристики й параметри похибки шукаються за відповідними методиками, що подаються у стандартах і базуються на статистиках випадкових величин чи послідовностей.

Похибка алгоритму визначається за характеристиками та параметрами похибок операцій на базі лінійної моделі алгоритму (з використанням принципів суперпозиції, однорідності) та його характеристик (передачі, вагових функцій тощо).

1.2. Р о б а с т н і с т ь . Робастністю у загальному називають властивість слабкої "реакції" алгоритму (наприклад, у межах його похибки) на випадкові, однократні відхилення значень вхідної величини чи параметрів (що інакше називають стійкістю "за Ляпуновим") чи їх неточні значення (що інакше називають чутливістю, або стійкістю "у малому" чи за Адроновим). Строгі означення стійкості та чутливості алгоритму даватимемо на базі таких означень для електричних чи цифрових кіл (схем).

1.3. Ш в и д к о д і я . За алгоритмом будується направлений граф, вершинами якого є операції, а гілки мають вагу часу виконання цих операцій. Вершини нумеровані за відношенням порядку виконання операцій. Для оцінки швидкодії шукатимемо всі шляхи на графі та оцінюватимемо їх величини, підсумовуючи ваги відповідних гілок у шляху. За оцінку швидкодії братимемо відповідні форми (лінійні чи квадратичні) довжин шляхів (у залежності від контексту).

1.4. С к л а д н і с т ь а л г о р и т м у . Поняття "алгоритм" уточнюється моделями: (а) — задачі та (б) — подання алгоритму. Оцінюватимемо алгоритм на базі цих моделей, визначаючи його складність. Складність алгоритму визначає його ефективність. Інтуїтивно простий алгоритм уявляється ефективним. Проте для числового подання складності потрібне її строге означення.

<u>Означення 18.</u> Складністю алгоритму α називається деяке число σ, отримане від обчислення функціоналу $\mathbf{M}(\alpha)=\sigma$, заданого на множині алгоритмів $A\ni\alpha$.

За цим означенням виходить, що функціонал та алгоритм мають означуватися на базі одної математичної моделі. Подібне означення можна сформулювати і для задачі.

Відповідні функціонали застосовують давно: наприклад, в числових методах розв’язування задач математичної фізики, обчислюючи кількість операцій (може певних) алгоритму, вибраного з множини подібних алгоритмів, розв’язування відповідної задачі, або обчислюючи тривалість розв’язування задач одного класу заданим алгоритмом. Отримані числа, зіставлені з аргументами відповідних функціоналів, називають сигнальними функціями.

Вважають, що складність алгоритму виражається натуральним числом. Отже, сигнальна функція ставить у відповідність кожному алгоритму розв’язування задачі натуральне число. Для кожної розв’язної (численої) задачі існує мінімально складний алгоритм для її розв’язування. Слушно розглядати цю мінімальну складність (алгоритму) як складність самої задачі. На множині таких задач існує відношення порядку (за складністю).

Складність фіксованої задачі частково впорядковує алгоритми. У цьому випадку потрібно оцінювати зверху і знизу дві сигнальні функції.

Сигнальну функцію для класу алгоритмів визначають шляхом випробування (тестування) алгоритму - представника класу, математичним аналізом його представлення. При цьому використовують асимптотичне наближення результату визначення сигнальної функції, яке позначають спеціальним виразом (наприклад, в нотації Ландау). Існують різні варіанти нотації складності (залежно від галузі, області, в якій цю функцію визначають).

ЛЕКЦІЯ 8

Тема: Оптимізація алгоритмів

1. Поняття оптимальності алгоритму

Раніше було встановлено, що за математичну модель явища вибирають математичний вираз — формулу, алгебричне рівняння, систему лінійних рівнянь, диференційне рівняння тощо. Для кожного з цих об’єктів можна виконати їх тотожні перетвори чи ізоморфні відображення і їх тотожні перетвори. Алгоритми вимірювань (інформування), керування будуються на базі цих моделей. З математичної точки зору алгоритми побудовані на базі однієї моделі однакові. Проте технічна реалізація цих алгоритмів порушує це. Виникає задача побудови критерію вибору технічно оптимального алгоритму з множини математично однакових.

Означення 19. Критерієм називається функціонал, який ставить у відповідність алгоритму число.

Очевидно, що такий функціонал повинен існувати, бути обчислюваним і означується (задається) на базі моделі алгоритму з оглядом на практичні потреби. Якщо функціонал опуклий, то існує його екстремум, що важливо для оптимізації. Квадратичні функціонали (та форми) опуклі, як і їх лінійні комбінації.

Посеред функціоналів вирізняються такі, що визначають час отримання результату (швидкодію), його точність і її залежність від точності даних та параметрів обчислень (робастність), кількість апаратних ресурсів, споживану енергію, вартість тощо. Лінійні комбінації цих характеристик також вибираються з практичних міркувань для вибору варіанту алгоритму.

3. Критерій оптимальності та обмеження

Критерій оптимальності алгоритму подається лінійною

$$\chi = \sum_k \beta_k \theta_k$$

("цінова" функція) чи квадратичною

$$\chi = \sum_k \beta_k \theta_k^2$$

("енергетична" функція) формами — зважені емпіричними чи евристичними коефіцієнтами β_k параметри θ_k характеристик алгоритму, $k = \overline{1, K}$, K — число всіх параметрів. Кожен параметр обчислюють за моделлю відповідної характеристики (складності, точності, швидкодії тощо). Важливою характеристикою критерію є його опуклість. Тоді існує глобальний його екстремум. Квадратична форма є опуклою. Вибір вигляду форми обгрунтовується евристично.

Обмеження задають технічні засоби реалізації алгоритму — енергетичні, габарити, фізичні фундаментальні тощо. Обмеження, як правило, мають вигляд системи нерівностей, побудованих на базі лінійних форм.

4. Розв’язування задачі оптимізації алгоритму.

Для постановки задачі оптимізації алгоритму потрібно обгрунтувати: вибір критерію оптимальності, вигляд технічних та фізичних обмежень засобів реалізації алгоритму, вибір початкової структури алгоритму, стратегію та метод пошуку оптимального алгоритму, спосіб визначення досягнення мети пошуку. Вибір початкового алгоритму, метод пошуку оптимального алгоритму, спосіб визначення зупинки пошуку залежить від вигляду критерію та обмежень. Загалом, це називають розв′язуванням задачі оптимізації (алгоритму). Такі задачі розв′язують на базі теорії дискретної оптимізації. При цьому необхідно мати множину алгоритмів. Її "генерують" за допомогою тотожних перетворів структури алгоритму методами дискретної математики, алгебри моделі задачі, зважаючи на відношення порядку. Формально оптимізація подається виразом

$$\operatorname*{argmin}_{\theta \subset \Theta} \chi ,$$

де θ — підмножина параметрів для вибраного алгоритму, Θ — множина всіх підмножин параметрів.

ЛІТЕРАТУРА

1. Грин Д., Кнут Д. Математические методы анализа алгоритмов.- М.: Мир, 1987.- 120 с.
2. Алферова. З.В. Теория алгоритмов.- М.: Статистика, 1973.- 164 с.
3. Кириллов А.А. Элементы теории представлений. - М.: Наука,1972.- 336 с.
4. Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов. – М.: Наука, 1984. – 224 с.
5. Драґан Я.П., Сікора Л.С., Яворський Б.І. Основи сучасної теорії стохастичних сигналів: енергетична концепція, математичний апарат, фізичне тлумачення. - Львів: ЕБТЕС, 1999.- 132 с.
6. Кузьмин И.В., Березюк И.Т., Фурманов К.К., Шаронов В.Б. Синтез вычислительных алгоритмов управления и контроля. – Л.: Техніка, 1975. – 248 с.
7. Яворський Б. І. Математичні основи радіоелектроніки/ В 3-х частинах. – Тернопіль: ТПІ, 1996. – 336 с.
8. Вирт Н. Алгоритмы + структуры данных = программы.— М.: Мир, 1985.-406 с.
9. Громов Г.Р. Очерки информационной технологии.— М.: ИнфАрт, 1992. – 336 с.
10. Сигорский В.П. Математический аппарат инженера.— К.: Техника, 1975.— 768 с.
11. Томович Р., Вукобратович М. Общая теория чувствительности.— М.: Сов. радио, 1972.—240 с.
12. Александров А.Г. Оптимальные и адаптивные системы.— М.: Высш. школа, 1989.— 263 с.
13. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации.— Киев: Наук. думка, 1988.— 472 с.
14. Капітонова Ю.В., Кривий С.Л., Летичевський О.А. та. ін. Основи дискретної математики.— Київ: Наук. думка, 2002.— 560 с.
15. Брукшир Дж. Гленн Введение в компютерные науки.— М.: Издательский дом "Вильямс", 2001.— 668 с.

ТИПОВІ КОНТРОЛЬНІ ПИТАННЯ ТА ЗАДАЧІ

Контрольні запитання

1.Характеристика евристичного поняття алгоритму.

2.Машина Поста. Система команд та технічні засоби.

3.Елементарна система керування.

4.Основні задачі технічних засобів в системах керування.

5.Поняття про інформацію. Одиниці вимірювання та засоби опрацювання інформації.

6.Поняття про математичні моделі, їх роль в інформаційних керівних системах.

7.Роль звичайних диференціальних рівнянь в інформаційних керівних системах.

8.Метод вербального подання алгоритму, його умови та обмеження.

9.Метод подання алгоритмів направленими графами.

10.Подання алгоритмів у вигляді блок-схем.

11.Подання алгоритму елементарними операторами.

12.Загальні властивості алгоритмів.

13.Складність алгоритмів.

14.Швидкодія алгоритму.

15.Еквівалентність алгоритмів.

16.Проблема алгоритмічної розв’язності.

17.Конвейєризація алгоритму.

18.Паралелення алгоритму.

19.Систолічні алгоритми.

20.Тотожні перетворення алгоритму.

21.Поняття оптимальності алгоритму. Методи оптимізації алгоритмів.

22.Стійкість алгоритму. Чутливість алгоритму до малих змін параметрів. Поняття про робастність.

23. Оцінювання характеристик та параметрів алгоритмів. Метрологія алгоритмів.

24.Роль та значення теорії алгоритмів для практики інформаційно керуючих систем.

25.Подання та представлення інформації. Алгоритмічні та інформаційні ресурси.

Задачі

1. Подати словесний алгоритм роботи елементарної системи керування.
2. Подати граф-схему алгоритму елементарної системи керування.
3. Подати блок-схему алгоритму елементарної системи керування.
4. Скласти перелік елементарних операторів системи керування.
5. Довести алгоритмічну розв’язність лінійних диференціальних рівнянь з постійними коефіцієнтами.

6. Подати словесний алгоритм розв'язку задачі знаходження найбільшого спільного дільника двох чисел (алгоритму Евкліда).
7. Подати граф-схему алгоритму Евкліда.
8. Подати блок-схему алгоритму Евкліда.
9. Скласти перелік елементарних операторів розв'язку задачі знаходження найбільшого спільного дільника двох чисел.
10. Довести алгоритмічну розв'язність задачі знаходження найбільшого спільного дільника.
11. Оцінити складність алгоритму Евкліда.
12. Оцінити складність алгоритму роботи елементарної системи керування.
13. Оцінити складність алгоритму розв'язування звичайного диференціального рівняння другого порядку з постійними коефіцієнтами.
14. Порівняти складність трансверсального та рекурсивного алгоритмів розв'язування звичайного диференціального рівняння другого порядку.
15. Оцінити час виконання алгоритму знаходження найбільшого спільного дільника двох чисел (в умовних одиницях).
16. Оцінити час виконання циклу керування алгоритму роботи елементарної системи керування (в умовних одиницях).
17. Оцінити час розв'язування звичайного диференціального рівняння першого порядку (в умовних одиницях).
18. Отримати алгоритм еквівалентний до алгоритму роботи елементарної системи керування.
19. Показати еквівалентність трансверсального та рекурсивного алгоритму.
20. Дослідити конвеєризацію та розпаралелення алгоритму розв'язування диференціального рівняння другого порядку.
21. Отримати алгоритм, еквівалентний до вже відомого алгоритму пошуку найбільшого спільного дільника двох чисел.
22. Отримати алгоритм, еквівалентний до алгоритму розв'язування диференціального рівняння другого порядку.
23. Подати алгоритм роботи елементарної системи керування.
24. Подати алгоритм розв'язку задачі знаходження найбільшого спільного дільника двох чисел (алгоритм Евкліда) командами машини Поста.
25. Подати алгоритм роботи інтегруючої ланки командами машини Поста.

Вказівки до розв'язування задач. Для розв'язування задач потрібно пам'ятати, що:

1. Задача розв'язується в рамках певної математичної моделі (множини об'єктів — чисел, функцій, операторів з заданими їх властивостями, які мають фізичну, технічну інтерпретацію).
2. Алгоритм будується, якщо метод розв'язування задачі відомий.
3. Алгоритм подається відповідними засобами його опису.
4. При поданні алгоритму враховуються властивості та можливості засобів для його реалізації.
5. При розв'язуванні задач незнайомі поняття, вирази, проблеми потрібно перш за все знайти у поданій літературі та своєму конспекті лекцій. Потім проконсультуватися з колегами, студентами, кваліфікованими знайомими. Якщо після цього задача не зрозуміла, то йти на консультацію з викладачем курсу.