

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ПОЛІТЕХНІЧНИЙ КОЛЕДЖ
ОДЕСЬКОГО НАЦІОНАЛЬНОГО ПОЛІТЕХНІЧНОГО УНІВЕРСИТЕТУ
ВІДДІЛЕННЯ КОМП'ЮТЕРНОЇ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Курс лекцій
з дисципліни

«Алгоритми та структури даних»

Зміст

1	Рекурсія	6
1.1	Вступ	6
1.2	Ключові терміни	6
1.3	Розширені теоретичні відомості	7
1.4	Приклади обчислень	8
1.5	Індивідуальне завдання	9
A	Правила оформлення звіту	10
A.1	Титульна сторінка лабораторної роботи	10
A.2	Приклади блок-схем	11

Перелік ілюстрацій

Перелік таблиць

Listings

1.1	Ітераційна функція	8
1.2	Рекурентна функція	9

Лабораторна робота № 1

Рекурсія

Мета роботи

1.1 Вступ

У математиці для вирішення переважної більшості завдань використовуються методи, які в кінцевому рахунку можуть бути зведені до одного з двох базових способів: ітерації або рекурсії.

Ітерація означає кількаразове повторення одних і тих же дій, яке після деякої кількості кроків приводить до бажаного результату. Характерним прикладом ітераційного способу вирішення завдання є методи послідовних наближень рішення нелінійних рівнянь, у тому числі метод дотичних, метод хорд і т.д.

$$f(x) = 0; x = \varphi(x); x_0 = a; x_n = \varphi(x_{n-1}), n = 1, 2, \dots, |x_k - x_{k-1}| > \epsilon \quad (1.1)$$

Рекурсія являє собою посилення при описі об'єкта, дії на описуваний об'єкт, дія. Рекурсія означає рішення задачі за допомогою відомості рішення до самого себе. При цьому обчислення залежать від інших, в не-якому сенсі більш простих (зазвичай менших) значень аргументу або аргументів завдання. Повністю аналогічні механізми використовуються в базовій теорії рекурсивних функцій, у методі математичної індукції, а також в рекурентних послідовностях.

Наприклад:

$$a_k = 2a_{k-1} + k, \forall k > 0, a_0 = 1 \quad (1.2)$$

Рекурсивний алгоритм - це алгоритм, в описі якого прямо або побічно міститься звернення до самого себе. У техніці процедурного програмування дане поняття поширюється на функцію, яка реалізує рішення окремого блоку завдання за допомогою виклику зі свого тіла інших функцій, в тому числі і себе самої. Якщо при цьому на черговому етапі роботи функція організовує звернення до самої себе, то така функція є рекурсивної.

1.2 Ключові терміни

База рекурсії - це тривіальний випадок, при якому рішення задачі очевидно, тобто не потрібно звернення функції до себе.

Глибина рекурсивних викликів - це найбільше одночасне кількість рекурсивних звернень функції, визначальне максимальну кількість шарів рекурсивного стека.

Декомпозиція - це вираження загального випадку через більш прості підзадачі зі зміненими параметрами.

Корінь повного дерева рекурсивних викликів - це вершина повного дерева рекурсії, відповідна початковому зверненню до функції.

Непряма (взаємна) рекурсія - це послідовність взаємних викликів декількох функцій, організована у вигляді циклічного замикання на тіло первісної функції, але з іншим набором параметрів.

Обсяг рекурсії - це характеристика складності рекурсивних обчислень для конкретного набору параметрів, що представляє собою кількість вершин повного рекурсивного дерева без одиниці.

Параметризація - це виділення з постановки задачі параметрів, які використовуються для опису умови задачі і рішення.

Повне дерево рекурсії - це граф, вершинами якого є набори фактичних параметрів при всіх викликах функції, починаючи з першого звернення до неї, а ребрами - пари таких наборів, відповідних взаємним викликам.

Пряма рекурсія - це безпосереднє звернення рекурсивної функції до себе, але з іншим набором вхідних даних.

Рекурсивна тріада - це етапи вирішення завдань рекурсивним методом.

Рекурсивна функція - це функція, яка у своєму тілі містить звернення до самої себе зі зміненим набором параметрів.

Рекурсивний алгоритм - це алгоритм, у визначенні якого міститься прямий або непрямий виклик цього ж алгоритму.

Рекурсія - це визначення об'єкта за допомогою посилання на себе.

1.3 Розширені теоретичні відомості

Рекурсивні алгоритми зазвичай виходять на основі математичної постановки завдання. Найважливіше при побудові рекурсивного алгоритму: побачити однакові дії на поточному та попередньому кроці обчислень (дій).

Виконавець рекурсивного алгоритму зводить невідоме до іншого невідомого, накопичуючи інформацію (прямий хід) і відкладаючи фактичні обчислення до моменту, коли виконається умова, що дозволяють безпосередньо обчислити шукане значення. Потім виконується зворотний хід рекурсії.

Основні переваги рекурсії:

- ◇ простота математичного формулювання;
- ◇ простота алгоритму і його реалізації

Основні недоліки рекурсії:

- ◇ додаткові витрати оперативної пам'яті;

- ◇ додаткові тимчасові витрати;
- ◇ можливий перехід складності в клас EXP.

За аналогією з математичною індукцією, на яку рекурсія трохи схожа, будь рекурсивна процедура повинна включати в себе базис і крок рекурсії.

Базис рекурсії - це пропозиція, що визначає якусь початкову ситуацію або ситуацію в момент припинення. Як правило, в цій пропозиції записується якийсь найпростіший випадок, при якому відповідь виходить відразу навіть без використання рекурсії. Так, у наведеній вище процедурі, яка описує предикат предок, базисом рекурсії є перше правило, в якому визначено, що найближчими предками людини є його батьки. Ця пропозиція часто містить умову, при виконанні якій відбувається вихід з рекурсії або відсікання.

Крок рекурсії - це правило, в тілі якого обов'язково міститься, в якості підділі, виклик обумовленого предиката. Якщо ми хочемо уникнути зациклення, який визначається предикат повинен викликатися не від тих же параметрів, які вказані в заголовку правила. Параметри повинні змінюватися на кожному кроці так, щоб в результаті або спрацював базис рекурсії, або умова виходу з рекурсії, розміщене в самому правилі.

1.4 Приклади обчислень

Приклад 1. Обчислення факторіала $P = n!$, n - ціле число

Зазвичай для обчислення факторіала цілого числа використовується ітераційний спосіб, заснований на багаторазовому домноженні величини, в якій накопичується результат, на черговий співмножник:

$$P_i := P_{i-1} \times i, i = 2, 3, \dots, k. \quad (1.3)$$

Лістинг 1.1 – Ітераційна функція

```
function factorial (k:integer):integer;
  var P,i:integer; { i - номер сомножителя, P - накапливаемый результат}
begin
  P:=1;
  for i:=1 to k do
    P:=P*i;
  factorial:=P;
end;
```

Це завдання можна вирішити і за допомогою рекурсії, базуючись на наступних міркуваннях:

$$k! = 1 \times 2 \times 3 \times \dots \times k = 1 \times 2 \times 3 \times \dots \times (k-1) \times k = (k-1)! \times k \quad (1.4)$$

Отже, $P(k) = k!$ можна визначити таким чином:

$$P(k) = \begin{cases} 1, & k = 1 \\ P(k-1) \times k, & k > 1 \end{cases} \quad (1.5)$$

Лістинг 1.2 – Рекурентна функція

```
function factorial1 (k:integer):integer ;
begin
  if k=1 then factorial1:=1
    else factorial1:= factorial1(k-1)*k
end;
```

1.5 Індивідуальне завдання

Завдання до лабораторної роботи

1. Вивчити теоретичний матеріал
2. Відповісти на контрольні запитання
3. Скласти звіт
4. Захистити роботу

Контрольні запитання

1. Що таке Internet? З яких структурних частин складається Internet?
2. Що таке IP-адреса?
3. Що таке доменне ім'я, з чого воно складається?
4. Який сервіс Internet перетворює IP-адреси в доменні імена і навпаки?
5. Яка служба займається розподіленням блоків IP-адрес?
6. Протокол HTTP. Рівень у моделі OSI, призначення.
7. Значення URI, URL, URN.
8. Мови web-програмування, які ви знаєте.
9. Веб-сервери, які ви знаєте.
10. Мережеві СКБД, які ви знаєте.

Додаток А
Правила оформлення звіту

А.1 Титульна сторінка лабораторної роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Звіт до лабораторної роботи №123
з дисципліни «Web-програмування»

Тема: «Основи мережі Internet»

Виконав
ст.групи хПР1

Пупкін А.А.

Перевірив
ст.викладач

Іванов Б.Б.

Херсон 2012

А.2 Приклади блок-схем

Правила виконання блок-схем задані наступними документами:

- ◇ ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения
- ◇ ГОСТ 19.002-80. Схемы алгоритмов и программ. Правила выполнения
- ◇ ГОСТ 19.003-80. Схемы алгоритмов и программ. Обозначения условные графические