

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
МАРІУПОЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**Кафедра** математичних методів та системного аналізу

**КОМПЛЕКС НАВЧАЛЬНО-МЕТОДИЧНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ  
ДИСЦИПЛІНИ**

Алгоритми та структури даних

---

Для спеціальності (напряму підготовки) 124 Системний аналіз

Автори: доцент Назаренко Н.В.  
Посада, прізвище та ініціали

Затверджено на засіданні кафедри протокол № 5 від «15» листопада 2016р.

**ЗМІСТ КОМПЛЕКСУ НАВЧАЛЬНО-МЕТОДИЧНОГО ЗАБЕЗПЕЧЕННЯ  
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
«АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ»**

- Навчальна програма дисципліни;
- Робоча програма навчальної дисципліни;
- Навчальний контент;
- Методичні рекомендації до проведення лабораторних робіт та організації СРС (в т.ч. завдання до індивідуальних проєктів);
- Засоби діагностики навчальних досягнень студентів;
- Інформаційні матеріали бібліотеки по забезпеченню навчальними підручниками (посібниками) з дисципліни.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Маріупольський державний університет**

---

**Алгоритми та структури даних**

(назва навчальної дисципліни)

**ПРОГРАМА**

**навчальної дисципліни**

**підготовки бакалавра**

(назва освітнього рівня / освітньо-кваліфікаційного рівня)

**спеціальності:** 6.040303 Системний аналіз

**(шифр за освітньою програмою ПП 08)**

**Маріуполь  
2016 рік**

**РОЗРОБЛЕНО ТА ВНЕСЕНО:** Маріупольський державний університет  
(повне найменування вищого навчального закладу)

**РОЗРОБНИКИ ПРОГРАМИ:** кандидат технічних наук, доцент кафедри математичних методів та системного аналізу Назаренко Н.В.

## ВСТУП

Програма вивчення нормативної навчальної дисципліни “Алгоритми та структури даних” складена відповідно до загальних компетентностей здобувачів ступеня бакалавра за спеціальністю «Системний аналіз».

**Предметом** вивчення навчальної дисципліни є технології складання алгоритмів та вибору типів структур, необхідних для вирішення поставлених задач фахового спрямування.

### **Міждисциплінарні зв'язки.**

Вивчення дисципліни “Алгоритми та структури даних” базується на знаннях, отриманих з курсу «Програмування та алгоритмічні мови». Матеріал цієї дисципліни використовується у подальшому вивченні дисциплін «Об'єктно-орієнтоване програмування» та «Технології програмування», а знання, набуті в процесі цього вивчення, використовуються під час майбутньої діяльності і професійного становлення висококваліфікованого фахівця.

Програма навчальної дисципліни складається з таких змістових модулів:

1. Аналіз алгоритмів.
2. Структури даних.
3. Алгоритми пошуку та сортування.

## **1. Мета та завдання навчальної дисципліни**

1.1. Метою викладання навчальної дисципліни “Алгоритми та структури даних” є формування у студентів системи знань в області алгоритмізації та структур даних, а також вмінь і навичок складання алгоритмів та вибору типів структур, необхідних для вирішення поставлених задач фахового спрямування.

1.2. Основними завданнями вивчення дисципліни “Алгоритми та структури даних” є: оволодіння основами алгоритмізації на рівні, достатньому для опрацювання задач системного аналізу, пов'язаних з подальшою практичною діяльністю фахівця в області моделювання об'єктів і процесів, напрацювання навичок самостійної роботи з науковою літературою, розглядання методів дослідження та розв'язання прикладних задач.

1.3. Згідно з вимогами освітньо-професійної програми студенти повинні:

### **знати :**

- теоретичні основи алгоритмізації і проектування програм;
- методи структурного програмування;
- розподіл оперативної пам'яті під статичні та динамічні об'єкти;
- засоби використання статичних та динамічних структур даних.

### **вміти :**

- використовувати, розробляти та досліджувати алгоритми розв'язування задач фахового спрямування;

- правильно вибрати та організовувати структури даних для конкретної задачі;
- використовувати рекурсивні структури даних, рекурсивні алгоритми;
- оцінювати складові ефективності алгоритмів функціонування комп'ютеризованих систем.

На вивчення навчальної дисципліни відводиться 90 годин/3 кредити ECTS або 120 годин/4 кредити ECTS згідно з навчальним планом.

## **2. Інформаційний обсяг навчальної дисципліни**

**Змістовий модуль 1. Аналіз алгоритмів.** Визначення інформації, алгоритму. Виконавці алгоритмів. Способи описання алгоритмів. Властивості алгоритмів. Поняття обчислювальної складності. Класи алгоритмів. Лінійні, розгалужені та циклічні алгоритми.

**Змістовий модуль 2. Структури даних.** Поняття структури даних. Рівні описування даних. Класифікація структур даних у програмах користувача та у пам'яті комп'ютера. Поняття структури даних (СД) типу масив, набір допустимих операцій. СД типу множина, запис (прямий декартовий добуток), таблиця, стек, черга, дек. СД типу вказівник. Статичні й динамічні змінні. СД типу лінійний однозв'язний список, циклічний лінійний список, двозв'язний лінійний список, багатозв'язний список. Хешування даних. Визначення дерева. Бінарне дерево. Подання дерев у зв'язаній пам'яті комп'ютера. Поняття графу. Подання графу у зв'язаній пам'яті комп'ютера. Алгоритми проходження дерев та графів.

**Змістовий модуль 3. Алгоритми пошуку та сортування.** Загальна класифікація алгоритмів пошуку. Лінійний пошук. Бінарний пошук елемента в масиві. Пошук методом Фібоначчі. М-блоковий пошук. Пошук у таблиці. Прямий пошук рядка. Сортування включенням, підрахунком, вибором, поділом (Хоара), злиттям. Метод Шелла. Обмінне сортування. Сортування за допомогою дерева. Пірамідальне сортування.

## **3. Рекомендована література.**

### **Базова**

1. Алгоритми і структура даних [Текст] : посібник / Н. Б. Шаховська, Р. О. Голошук. - Львів : Магнолія 2006, 2014. - 215 с. - (Серія "Комп'ютинг").
2. Вирт Никлаус. Алгоритмы и структуры данных : пер. с англ. / Н. Вирт. — 2-е изд., испр. — СПб. : Невский Диалект, 2001. — 351 с.
3. Математична логіка. Основи теорії алгоритмів [Текст] : навчальний посібник / С. С. Шкільняк. - К. : ДП "Видавничий дім "Персонал", 2009. - 280 с.

### Допоміжна

4. Алгоритмы: вводный курс / Томас Х. Кормен : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2014. — 208 с.: ил.
5. Ахо Альфред В., Хопкрофт Джон, Ульман Джеффри Д. Структуры данных и алгоритмы.: Пер. с англ.: Уч.пос. – М.: Издательский дом «Вильямс», 2003. – 384 с.
6. Козак Л.І. Основи програмування : навчальний посібник / Л. І. Козак, І. В. Костюк, С. П. Стасевич. - Львів : Новий Світ-2000, 2012.
7. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ. - М.: Издательский дом «Вильямс», 2005.
8. Красиков И.В. Алгоритмы. Просто как дважды два./ И.В.Красиков, Е.И. Красикова. – М.: Эксмо, 2007. – 256 с.
9. Х. Ричард, К.Лоуренс. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста. Пер. с англ./ Х. Ричард, К.Лоуренс. – К.: Издательство «ДиаСофт», 2001.-736 с.

### Джерела Internet

1. Алгоритмы и структуры данных. Лекториум  
<https://www.lektorium.tv/lecture/13343>
  2. Алгоритмы и структуры данных. Видеолекции Технопарка.  
<http://habrahabr.ru/company/abbyy/blog/251561/>
  3. Типичный программист. Алгоритмы и структуры данных для начинающих.  
<http://tproger.ru/tag/for-beginners/>
- 4. Форма підсумкового контролю успішності навчання – 2 або 3 семестр (згідно з навчальним планом), форма контролю – залік.**

### **5. Засоби діагностики успішності навчання.**

Рекомендується застосовувати такі засоби діагностики рівня підготовки студентів: поточне оцінювання результатів виконання практичних та лабораторних робіт; оцінювання самостійної роботи студентів; оцінка за індивідуальне навчально-дослідне завдання; модульна контрольна робота в письмовій формі.

Звертається увага на уміння студентів чітко формулювати висновки і пропозиції, витримувати логічну послідовність та на грамотність мови, правильність виконання завдань на персональному комп'ютері.

# МАРІУПОЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра \_\_\_\_\_ математичних методів та системного аналізу \_\_\_\_\_

“ЗАТВЕРДЖУЮ”  
Завідувач кафедри математичних  
методів та системного аналізу  
\_\_\_\_\_ Коляда Ю.Є.  
“ 30 ” \_\_\_\_\_ 2016 року



## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### ПП8 Алгоритми та структури даних

(шифр і назва навчальної дисципліни)

спеціальність \_\_\_\_\_ 124 Системний аналіз \_\_\_\_\_  
(код та найменування спеціальності)

спеціалізація \_\_\_\_\_  
(назва спеціалізації)

факультет \_\_\_\_\_ економіко-правовий \_\_\_\_\_  
(назва факультету)

2016 – 2017 рік



Робоча програма Алгоритми та структури даних для студентів  
(назва навчальної дисципліни)

Спеціальністю (напрямом підготовки) 124 Системний аналіз

Розробники:

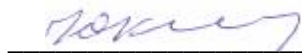
Назаренко Наталя Вікторівна, доцент кафедри математичних методів та системного аналізу,  
к.т.н., доцент

(вказати авторів, їхні посади, наукові ступені та вчені звання)

Робоча програма затверджена на засіданні  
кафедри математичних методів та системного аналізу

Протокол від “29” серпня 2016 року, № 1

Завідувач кафедри математичних методів та системного аналізу



(підпис)

(Коляда Ю.С.)

(прізвище та ініціали)

“ 30 ” 08 2016 року

© Назаренко Н.В., 2016 рік

© МДУ, 2016 рік

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітній рівень / освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 4	Галузь знань <u>12 Інформаційні технології</u> (шифр і назва)	за вибором	
Модулів – 2	Спеціальність(напрям підготовки): <u>124 Системний аналіз</u> (код та найменування спеціальності)	Рік підготовки:	
Змістових модулів – 3		1-й	-й
Індивідуальне науково-дослідне завдання <u>складання алгоритмів для розв’язання поставлених завдань</u> (назва)		Семестр	
Загальна кількість годин - 120		2-й	-й
		Лекції	
Тижневих годин для денної форми навчання: аудиторних – 2 самостійної роботи студента – 3	Освітній рівень / освітньо-кваліфікаційний рівень: <u>бакалавр</u>	20 год.	год.
		Практичні, семінарські	
		12 год.	год.
		Лабораторні	
		16 год.	год.
		Самостійна робота	
		68 год.	год.
		Індивідуальні завдання:	
		4 год.	
		Вид контролю:	
	залік		

### Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 47% : 53%

## **2. Мета та завдання навчальної дисципліни**

**Мета:** формування системи знань в області алгоритмізації та структур даних, а також вмінь і навичок складання алгоритмів та вибору типів структур, необхідних для вирішення поставлених задач фахового спрямування.

**Завдання:** оволодіння основами алгоритмізації на рівні, достатньому для опрацювання задач системного аналізу, пов'язаних з подальшою практичною діяльністю фахівця в області моделювання об'єктів і процесів, напрацювання навичок самостійної роботи з науковою літературою, розглядання методів дослідження та розв'язання прикладних задач.

У результаті вивчення навчальної дисципліни студент повинен

**знати:**

- теоретичні основи алгоритмізації і проектування програм;
- методи структурного програмування;
- розподіл оперативної пам'яті під статичні та динамічні об'єкти;
- засоби використання статичних та динамічних структур даних.

**вміти:**

- & використовувати, розробляти та досліджувати алгоритми розв'язування задач фахового спрямування;
- & правильно вибрати та організовувати структури даних для конкретної задачі;
- & використовувати рекурсивні структури даних, рекурсивні алгоритми;
- & оцінювати складові ефективності алгоритмів функціонування комп'ютеризованих систем.

## **Програма навчальної дисципліни**

### **Змістовий модуль 1. Аналіз алгоритмів**

#### **ТЕМА 1. Базові поняття теорії алгоритмізації.**

Визначення інформації, алгоритму. Виконавці алгоритмів. Способи описання алгоритмів. Властивості алгоритмів. Поняття обчислювальної складності. Класи алгоритмів. Лінійні, розгалужені та циклічні алгоритми.

### **Змістовий модуль 2. Структури даних**

#### **ТЕМА 1. Поняття структури даних. Структурні та лінійні типи даних.**

Поняття структури даних. Рівні описування даних. Класифікація структур даних у програмах користувача та у пам'яті комп'ютера. Основні види складених типів даних. Структури даних у пам'яті комп'ютера.

Поняття структури даних (СД) типу масив, набір допустимих операцій. Дескриптор СД типу масив. Ефективність масивів. СД типу множина, запис (прямий декартовий добуток), таблиця, стек, черга, дек.

СД типу вказівник. Статичні й динамічні змінні. СД типу лінійний однозв'язний список, циклічний лінійний список, двозв'язний лінійний список, багатозв'язний список.

## **ТЕМА 2. Хешування даних.**

Поняття хеш-функції. Алгоритми хешування. Динамічне хешування. Методи розв'язування колізій. Переповнення таблиці і ре хешування. Оцінювання якості хеш-функції.

## **ТЕМА 3. Нелінійні структури даних.**

Визначення дерева. Бінарне дерево. Подання дерев у зв'язаній пам'яті комп'ютера. Алгоритми проходження дерев углиб і вшир. Подання дерев у вигляді бінарних. Види бінарних дерев.

Поняття графу. Подання графу у зв'язаній пам'яті комп'ютера. Алгоритми проходження графу. Топологічне сортування. Пошук мостів. Найкоротша відстань між вершинами.

## **Змістовий модуль 3. Алгоритми пошуку та сортування**

### **ТЕМА 1. Алгоритми пошуку.**

Загальна класифікація алгоритмів пошуку. Лінійний пошук. Бінарний пошук елемента в масиві. Пошук методом Фібоначчі. М-блоковий пошук. Пошук у таблиці. Прямий пошук рядка. Алгоритми Ахо-Корасик, Моріса-Прата, Кнута, Рабіна-Карпа, Боуера і Мура, Хорспула. Порівняння методів пошуку.

### **ТЕМА 2. Алгоритми сортування.**

Сортування включенням, підрахунком, вибором, поділом (Хоара), злиттям. Метод Шелла. Обмінне сортування. Сортування за допомогою дерева. Пірамідальне сортування.

Методи зовнішнього сортування: пряме злиття, природне злиття, збалансоване багато шляхове злиття, багатофазне сортування.

Поняття та приклади жадібних алгоритмів.

#### 4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Модуль 1</b>												
<b>Змістовий модуль 1. Аналіз алгоритмів</b>												
Тема 1. Базові поняття теорії алгоритмізації	14	2	2			10						
<b>Разом за змістовим модулем 1</b>	<b>14</b>	<b>2</b>	<b>2</b>			<b>10</b>						
<b>Змістовий модуль 2. Структури даних</b>												
Тема 1. Поняття структури даних. Структурні та лінійні типи даних	19	4	2	2		11						
Тема 2. Хешування даних	13	2	2	2		7						
Тема 3. Нелінійні структури даних	17	4	2	2		9						
<b>Разом за змістовим модулем 2</b>	<b>49</b>	<b>10</b>	<b>6</b>	<b>6</b>		<b>27</b>						
<b>Змістовий модуль 3. Алгоритми пошуку та сортування</b>												
Тема 1. Алгоритми пошуку	23	4	2	6		11						
Тема 2. Алгоритми сортування	22	4	2	4		12						
<b>Разом за змістовим модулем 3</b>	<b>45</b>	<b>8</b>	<b>4</b>	<b>10</b>		<b>23</b>						
<b>Усього годин</b>	<b>108</b>	<b>20</b>	<b>12</b>	<b>16</b>		<b>60</b>						
<b>Модуль 2</b>												
ІНДЗ	12				4	8						
<b>Усього годин</b>	<b>120</b>	<b>20</b>	<b>12</b>	<b>16</b>	<b>4</b>	<b>68</b>						

## 5. Теми семінарських занять

## 6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1.	Лінійні алгоритми.	2
2.	Розгалужені алгоритми.	2
3.	Циклічні алгоритми.	2
4.	Алгоритми пошуку.	2
5.	Алгоритми обробки масивів.	2
6.	Алгоритми сортування масивів.	2

## 7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1.	Розробка лінійних алгоритмів.	2
2.	Розробка розгалужених алгоритмів.	2
3.	Розробка циклічних алгоритмів.	2
4.	Алгоритми пошуку.	2
5.	Алгоритми обробки масивів.	4
6.	Алгоритми сортування масивів.	2
7.	Модульна контрольна робота	2

## 8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1.	Базові поняття теорії алгоритмізації	10
2.	Поняття структури даних. Структурні та лінійні типи даних	11
3.	Хешування даних	7
4.	Нелінійні структури даних	9
5.	Алгоритми пошуку	11
6.	Алгоритми сортування	12
7.	Підготовка індивідуального проекту	8

## 9. Індивідуальні завдання

Скласти алгоритми для розв'язання поставлених в індивідуальному варіанті задач.

## 10. Методи навчання

При проведенні лекційних занять з курсу «Алгоритми і структури даних» доцільно використовувати словесні методи навчання: пояснення, розповідь, бесіда, навчальна дискусія з поєднанням наочних методів навчання (ілюстрування, демонстрування, складання алгоритмів біля дошки) та використанням сучасних інтерактивних засобів навчання (презентацій, електронних гіпертекстових документів, мультимедійних розробок).

## 11. Методи контролю

Для контролю засвоєння матеріалу дисципліни «Алгоритми і структури даних» рекомендується використовувати такі методи і форми контролю: поточне оцінювання результатів виконання лабораторних робіт; оцінювання самостійної роботи студентів; оцінка за індивідуальне навчально-дослідне завдання; модульна контрольна робота у письмовій формі.

## 12. Розподіл балів, які отримують студенти

Модуль 1							МКР	Модуль2 (проект)	Сума
поточне тестування і самостійна робота									
ЗМ1	ЗМ2			ЗМ3		ЗМ1-3			
T1	T1	T2	T3	T1	T2	CPT	12	15	100
8	9	9	10	12	10	15			

CPT – теоретична самостійна робота

МКР – модульна контрольна робота

## Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання

0-34	Ф	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни
------	---	---	---

### 13. Методичне забезпечення

Опорний конспект лекцій; ілюстративні матеріали; Інтернет сайти; методичні вказівки до лабораторних робіт.

### 14. Рекомендована література

#### Базова

10. Алгоритми і структура даних [Текст] : посібник / Н. Б. Шаховська, Р. О. Голощук. - Львів : Магнолія 2006, 2014. - 215 с. - (Серія "Комп'ютинг").
11. Вирт Никлаус. Алгоритмы и структуры данных : пер. с англ. / Н. Вирт. — 2-е изд., испр. — СПб. : Невский Диалект, 2001. — 351 с.
12. Математична логіка. Основи теорії алгоритмів [Текст] : навчальний посібник / С. С. Шкільняк. - К. : ДП "Видавничий дім "Персонал", 2009. - 280 с.

#### Допоміжна

13. Алгоритмы: вводный курс / Томас Х. Кормен : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2014. — 208 с.: ил.
14. Ахо Альфред В., Хопкрофт Джон, Ульман Джеффри Д. Структуры данных и алгоритмы.: Пер. с англ.: Уч.пос. – М.: Издательский дом «Вильямс», 2003. – 384 с.
15. Козак Л.І. Основи програмування : навчальний посібник / Л. І. Козак, І. В. Костюк, С. П. Стасевич. - Львів : Новий Світ-2000, 2012.
16. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ. - М.: Издательский дом «Вильямс», 2005.
17. Красиков И.В. Алгоритмы. Просто как дважды два./ И.В.Красиков, Е.И. Красикова. – М.: Эксмо, 2007. – 256 с.
18. Х. Ричард, К.Лоуренс. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста. Пер. с англ./ Х. Ричард, К.Лоуренс. – К.: Издательство «ДиаСофт», 2001.-736 с.

### 15. Інформаційні ресурси

1. Алгоритмы и структуры данных. Лекториум  
<https://www.lektorium.tv/lecture/13343>
2. Алгоритмы и структуры данных. Видеолекции Технопарка.  
<http://habrahabr.ru/company/abbyy/blog/251561/>
3. Типичный программист. Алгоритмы и структуры данных для начинающих.  
<http://tproger.ru/tag/for-beginners/>



Навчальний контент  
з дисципліни «Алгоритми та структури даних»  
для студентів спеціальності «Системний аналіз»

## АЛГОРИТМ І ЙОГО ВЛАСТИВОСТІ

1. Поняття програми. Етапи, необхідні для вирішення завдань на ЕОМ
2. Алгоритм і його властивості
3. Етапи розробки і аналізу алгоритмів
4. Основи аналізу ефективності алгоритмів
5. Класи алгоритмів
6. Способи подання алгоритмів
7. Шаблони для подання алгоритму в графічному вигляді
8. Основні види процесів алгоритмізації
9. Похибки алгоритмів
10. Мови програмування

Загальним для всієї обчислювальної техніки є те, що для вирішення задач на ній необхідно скласти програму. Програма для ЕОМ - це запис алгоритмів розв'язання задачі у вигляді послідовності команд, інструкцій, які повинні бути виконані. Для вирішення завдань на ЕОМ необхідно виконати наступні основні етапи:

1. Постановка завдання.
2. Математичний опис задачі.
3. Розробка алгоритму рішення задач.
4. Програмування.
5. Тестування і налагодження програми.
6. Автоматизоване рішення задачі з використанням ЕОМ і аналіз отриманих результатів.

Під алгоритмом розуміється кінцева послідовність точно визначених дій (команд), які виконуються над вхідними даними і призводять до вирішення поставленого завдання.

Алгоритм повинен мати наступні властивості:

1. Дискретність (або кінцевість).
2. Визначення.
3. Результативність.
4. Правильність.
5. Масовість.
6. Ефективність.
7. Формальність.

Тимчасова ефективність є індикатором швидкості роботи алгоритму:

- оцінюється за кількістю основних операцій, які повинен виконати алгоритм при обробці вхідних даних розміру  $n$ ;
- важливий порядок зростання часу виконання алгоритму в залежності від  $n$ .

Просторова ефективність показує, скільки додаткової оперативної пам'яті потрібно для роботи алгоритму:

- ефективність оцінюють в найгіршому, найкращому і середньому випадках;
- види аналізу: математичний і емпіричний

При аналізі ефективності алгоритмів основна увага має бути зосереджена на порядку зростання кількості базових операцій алгоритму. Для того, щоб можна було порівнювати між собою ці порядки зростання і класифікувати їх, введений ряд умовних позначень. Основні з них -  $O$  (прописна "О"), і  $\Theta$  (прописна грецька "тета").

### **Рекомендована література.**

Базова: 1, 2, 3.

Додаткова: 4, 7, 8.

### **ПОНЯТТЯ СТРУКТУРИ ДАНИХ**

1. Основні визначення
2. Класифікація структур даних
3. Статистичні структури даних

Структури даних - це сукупність елементів даних і відносин між ними. При цьому під елементами даних може трактуватись як просте дане, так і структура даних. Під відносинами між даними розуміють функціональні зв'язки між ними і показники на те, де знаходяться ці дані.

Елемент відносин - це сукупність всіх зв'язків елемента з іншими елементами даних, що розглядається в структурі.

Пам'ять машини складається з мільйонів тригерів, які обробляють інформацію, що надходить. Ми, заносючи інформацію в комп'ютер, представляємо її в якомусь вигляді, який на наш погляд впорядковує дані і надає їм сенс. Машина відводить поле для інформації, що надходить і задає їй якусь адресу. Таким чином виходить, що ми обробляємо дані на логічному рівні, як би абстрактно, а машина робить це на фізичному рівні, тобто, можна виділити три рівні опису даних:

- 1) абстрактний (математичний рівень);
- 2) логічний рівень - представлення структури даних на будь-якій мові програмування;
- 3) фізичний рівень - відображення в пам'яті комп'ютера інформаційного об'єкта відповідно до логічного опису.

Структури даних класифікуються:

1. За пов'язаності даних в структурі:
  - якщо дані в структурі пов'язані дуже слабо, то такі структури називаються непов'язаними (вектор, масив, рядки, стеки);
  - якщо дані в структурі пов'язані, то такі структури називаються пов'язаними (пов'язані списки);
2. За мінливості структури в часі або в процесі виконання програми:
  - статичні структури - структури, незмінні до кінця виконання програми

(записи, масиви, рядки, вектори);

- напівстатичні структури (стеки, деки, черги);
- динамічні структури - відбувається повна зміна при виконанні програми (списки).

3. За впорядкованості структури:

- лінійні (вектори, масиви, стеки, деки, записи);
- нелінійні (багатозв'язкові списки, деревовидні структури, графи).

Найбільш важливою характеристикою є мінливість структури в часі.

Найпростіша статична структура - це вектор. Вектор - це чисто лінійна упорядкована структура, де відношення між її елементами є строго виражена послідовність елементів структури.

У загальному випадку елемент масиву - це є елемент вектора, який сам по собі теж є елементом структури.

Операції над масивами:

- 1) операція доступу;
- 2) операція присвоювання;
- 3) операція ініціалізації (визначення початкових умов).

### **Рекомендована література.**

Базова: 1, 2.

Додаткова: 5, 6, 9.

## **СТАТИЧНІ І НАПІВСТАТИЧНІ СТРУКТУРИ ДАНИХ**

1. Записи
2. Таблиці
3. Списки
4. Стеки
5. Алгоритми основних операцій зі стеком

Запис являє собою структуру даних послідовного типу, де елементи структури розташовані один за іншим як у логічному, так і у фізичному поданні. Запис передбачає безліч елементів різного типу. Елементи даних у запису часто називають полями запису.

Логічна структура запису може бути представлена як в графічному вигляді, так і в табличному.

Елемент запису може включати в себе запис. В цьому випадку виникає складна ієрархічна структура даних.

Операції над записами:

1. Читання вмісту поля запису.
2. Занесення інформації в поле запису.
3. Всі операції, які дозволяються над полем запису відповідного типу.

Таблиця - це кінцевий набір записів.

Операції з таблицями:

1. Пошук запису по заданому ключу.
2. Занесення нового запису в таблицю.

Ключ - це ідентифікатор запису. Для зберігання цього ідентифікатора відводиться спеціальне поле.

Складний ключ - ключ, що містить більше двох полів.

До напівстатичних структур даних відносять стеки, деки і черги. Перш, ніж почати докладне вивчення даних структур, розглянемо одну більш загальну структуру, яка називається списком.

Списки - це набір пов'язаних елементів даних, які в загальному випадку можуть бути різного типу.

Приклад списку:

$E_1, E_2, \dots, E_n, \dots n > 1$  і не зафіксовано.

Кількість елементів списку може змінюватися в процесі виконання програми.

Розрізняють 2 види списків:

- 1) незв'язні;
- 2) зв'язні.

У незв'язних списках зв'язок між елементами даних виражений неявно. У зв'язаних списках в елемент даних заноситься покажчик зв'язку з подальшим або попереднім елементом списку.

Стеки, деки і черги - це незв'язні списки. Крім того, вони відносяться до послідовних списків, в яких неявний зв'язок відображається їх послідовністю.

Черга виду LIFO (Last In First Out - Останнім прийшов, першим пішов), при якій на обслуговування першим вибирається той елемент черги, який надійшов в неї останнім, називається стеком. Це одна з найбільш вживаних структур даних, яка виявляється дуже зручною при вирішенні різних завдань.

Операції над стеками:

1. Занесення елемента в стек.
2. Вибірка елемента з стека.
3. Визначення порожнечі стека.
4. Читання елемента без його вибірки зі стека.
5. Визначення переповнення стека (для напівстатичних структур).

### **Рекомендована література.**

Базова: 1, 2.

Додаткова: 5, 6, 9.

## НАПІВСТАТИЧНІ СТРУКТУРИ

1. Черга
2. Операції над чергою
2. Деки

У програмуванні є структура даних, яка називається чергою. Ця структура даних використовується, наприклад, для моделювання реальних черг з метою визначення їх характеристик при даному законі надходження замовлень і дисципліни їх обслуговування. За своєю суттю черга, реалізована на масиві, є напівстатичною структурою - з плином часу і довжина черги, і склад можуть змінюватися.

На малюнку 1 наведена чергу, містить чотири елементи - А, В, С і D. Елемент А розташований на початку черги, а елемент D - в її кінці. Елементи можуть бути видалені тільки з початку черги, тобто перший об'єкт, що був поміщений в чергу, буде видалений першим. З цієї причини чергу часто називають списком, організованим за принципом «перший розміщений першим видаляється» на протипагу принципу стекової організації - «останній розміщений першим видаляється».

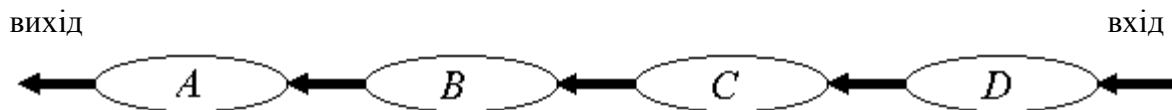


Рис. 1. Приклад організації черги

Дек (від англ. Deq - double ended queue, тобто черга з двома кінцями) - це такий послідовний список, в якому як включення, так і виключення елементів може здійснюватися з будь-якого з двох кінців списку. Окремий випадок дека - дек з обмеженим входом і дек з обмеженим виходом. Логічна і фізична структури дека аналогічні логічній і фізичній структурі кільцевої FIFO-черги. Однак, стосовно деку доцільно говорити не про початок і кінець, а про лівий та правий край.

### Рекомендована література.

Базова: 1, 2.  
Додаткова: 5, 6, 9.

## ДИНАМІЧНІ СТРУКТУРИ ДАНИХ

1. Загальні поняття про динамічні структури. Зв'язані списки
2. Однозв'язні списки
3. Двозв'язні списки
4. Найпростіші операції над однозв'язаними списками

Використання при програмуванні тільки статичних об'єктів може викликати певні труднощі, особливо з точки зору отримання ефективної машинної програми. Справа в тому, що іноді ми заздалегідь не знаємо не тільки розміру значення того чи іншого програмного об'єкта, але також і того, чи буде існувати цей об'єкт чи ні.

Такого роду програмні об'єкти, які виникають уже в процесі виконання програми або розмір значень яких визначається при виконанні програми, називаються динамічними об'єктами.

Динамічні структури даних мають дві особливості:

1. Заздалегідь не визначена кількість елементів в структурі.
2. Елементи динамічних структур фізично не мають жорсткої лінійної впорядкованості. Вони можуть бути розкидані по пам'яті.

Показчики містять номер комірки пам'яті, з якої починається відповідний елемент структури.

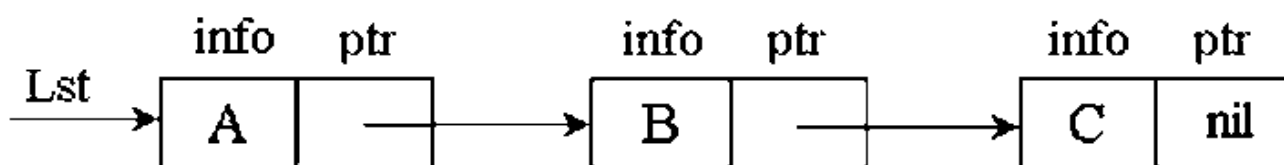
З точки зору логічного представлення розрізняють лінійні і нелінійні списки.

До лінійних списків відносяться однозв'язні і двозв'язні списки. До нелінійних - багатозв'язні.

Приклад в загальному випадку являє собою інформаційне поле і одне або кілька полів показників.

Дані динамічні структури найбільш поширені.

Елемент одинзв'язного списку містить, як мінімум, два поля: інформаційне поле (info) і поле показника (ptr).



Особливістю показника є те, що він дає тільки адресу наступного елемента списку. Поле показника останнього елемента в списку є порожнім (NIL). LST - показник на початок списку. Список може бути порожнім, тоді LST буде дорівнювати NIL. Доступ до елемента списку здійснюється тільки від його початку, тобто зворотного зв'язку в цьому списку немає.

При подальшому вивченні однозв'язних списків ми будемо використовувати наступну термінологію:

p - показник

node (p) - вузол, на який посилається вказівник p (при цьому неважливо в яке місце зображення елемента (вузла) списку він спрямований на малюнку)

ptr (p) - посилання на наступний елемент вузла node (p)

ptr (ptr (p)) - посилання подальшого для node (p) вузла на наступний для нього елемент.

### Рекомендована література.

Базова: 1, 2.

Додаткова: 5, 6, 9.

### НЕЛІНІЙНІ ПОВ'ЯЗАНІ СТРУКТУРИ

1. Нелінійні списки
2. Рекурсивні структури даних
3. Дерева

#### 4. Бінарні дерева

Двозв'язний список може бути нелінійною структурою даних, якщо другі покажчики задають довільний порядок проходження елементів (рис.1).

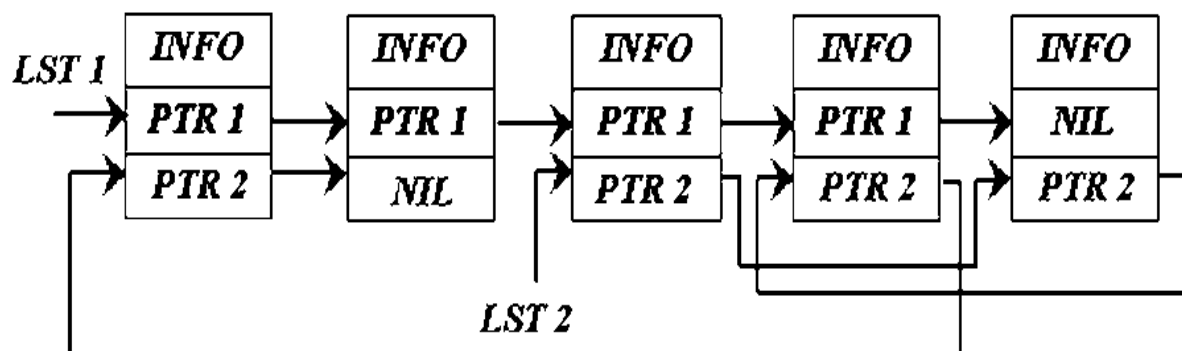


Рис. 1 – Приклад двозв'язного списку у вигляді нелінійної структури даних

LST1 - покажчик на початок першого списку (орієнтованого покажчиками PTR1). Він лінійний і складається з 5-ти елементів.

Другий список утворений з цих же самих елементів, але в довільній послідовності. Початком 2-го списку є 3-й елемент, а кінцем 2-ий елемент.

У загальному випадку елемент облікової структури може містити як завгодно багато покажчиків, тобто може вказувати на будь-яку кількість елементів.

Можна виділити 3 ознаки відмінності нелінійної структури:

- 1) Будь-який елемент структури може посилатися на будь-яке число інших елементів структури, тобто може мати будь-яке число полів-покажчиків.
- 2) На даний елемент структури може посилатися будь-яке число інших елементів цієї структури.
- 3) Посилання можуть мати вагу, тобто мається на увазі ієрархія посилань.

Приклад моделювання за допомогою нелінійного списку.

Нехай є дискретна система, в графі стану якої вузли - це стан, а ребра - переходи зі стану в стан (рис.2).

Вхідний сигнал в систему це X. Реакцією на вхідний сигнал є вироблення вихідного сигналу Y і перехід у відповідний стан.

Граф стану дискретної системи можна представити у вигляді двозв'язного нелінійного списку. При цьому в інформаційних полях повинна записуватися інформація про стани системи і ребра. Покажчики елементів повинні формувати логічні ребра системи.

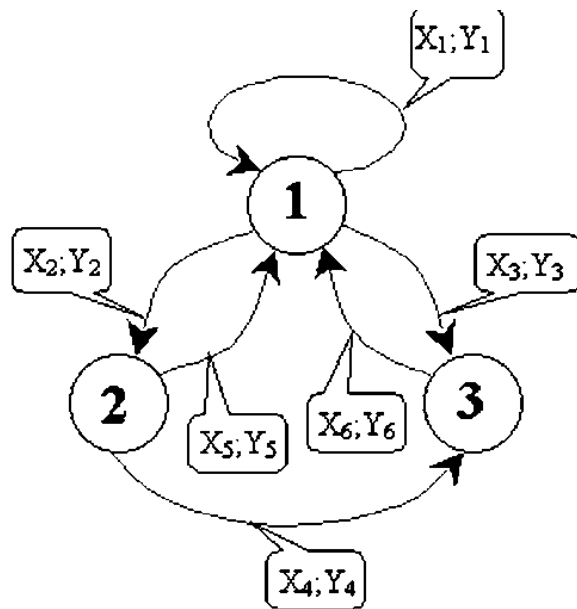


Рис.2 – Приклад дискретної системи у вигляді графа

Реалізацію графа у вигляді нелінійного списку можна представити у наступному вигляді (рис.3):

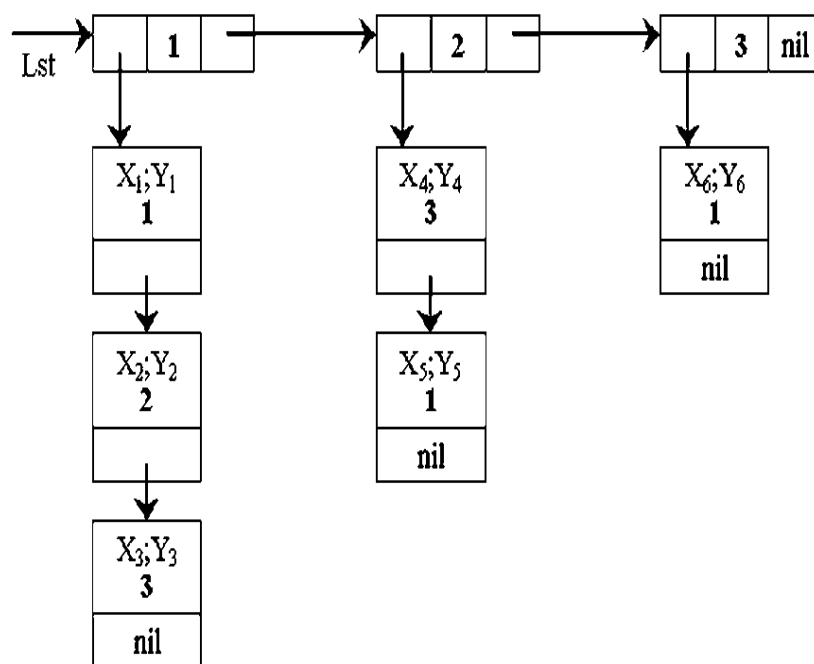


Рис.3 – Реалізація графа у вигляді нелінійного списку

Дерево - нелінійна пов'язана структура даних

Дерево характеризується наступними ознаками:

- Дерево має один елемент, на який немає посилань від інших елементів. Цей елемент називається коренем дерева;

- В дереві можна звернутися до будь-якого елементу шляхом проходження кінцевого числа посилань (покажчиків);

- Кожен елемент дерева пов'язаний тільки з одним попереднім елементом.

Будь-який вузол дерева може бути проміжним або термінальним (листом). Характерною особливістю термінального вузла є відсутність гілок.



## Рекомендована література.

Базова: 1, 2.

Додаткова: 5, 6, 9.

## КЛАСИФІКАЦІЯ АЛГОРИТМІВ ПОШУКУ. МЕТОДИ ОПТИМІЗАЦІЇ ПОШУКУ

1. Основні визначення
2. Послідовний пошук
3. Послідовний пошук в однозв'язному списку
4. Ефективність послідовного пошуку
5. Індексно-послідовний пошук і його ефективність
6. Методи оптимізації пошуку

Пошук є однією з основних операцій при обробці інформації в ЕОМ. Його призначення - по заданому аргументу знайти серед масиву даних ті дані, які відповідають цьому аргументу.

Набір даних (будь-яких) будемо називати таблицею або файлом. Будь-яке дане (або елемент структури) відрізняється якоюсь ознакою від інших даних. Ця ознака називається ключем. Ключ може бути унікальним, тобто в таблиці існує тільки одне дане з цим ключем. Такий унікальний ключ називається первинним.

Вторинний ключ в одній таблиці може повторюватися, але по ньому теж можна організувати пошук. Ключі даних можуть бути зібрані в одному місці (в іншій таблиці) або являти собою запис, в якому одне з полів - це ключ. Ключі, які виділені з таблиці даних і організовані в свій файл, називаються зовнішніми ключами. Якщо ключ знаходиться в запису, то він називається внутрішнім.

Пошуком по заданому аргументу називається алгоритм, який визначає відповідність ключа з заданим аргументом. Результатом роботи алгоритму пошуку може бути знаходження цього даного або відсутність його в таблиці.

У разі знаходження ключа (і інформації йому відповідної) можливі дві операції:

- обробка знайденої інформації;
- видалення знайденої інформації.

У разі відсутності ключа можливі теж дві операції:

- індикація того, що ключа немає;
- вставка ключа (і інформації) в таблицю.

Всі методи пошуку можна розділити на статичні і динамічні. При статичному пошуку масив значень не змінюється під час роботи алгоритму. При динамічному пошуку масив може перебудовуватися або змінювати розмірність.

*Ефективність послідовного пошуку в масиві*

Число порівнянь будемо позначати  $C$

$C_{\min} = 1$ ,  $C_{\max} = n$ .

Якщо дані розташовані рівновипадково у всіх комірках масиву, то

$C_{\text{ср}} \approx (n + 1)/2$ .

При індексно-послідовному пошуку організовується дві таблиці: таблиця даних зі своїми ключами - упорядкованих за зростанням, і таблиця індексів, яка теж складається з джерел інформації, але ці ключі взяті з основної таблиці через певний інтервал.



Порядок ефективності індексного-послідовного пошуку

У порівнянні з послідовним, ефективність індексного-послідовного пошуку набагато вище, особливо для великих масивів, але не слід забувати, що послідовність елементів повинна бути впорядкована.

### **Рекомендована література.**

Базова: 1, 2.

Додаткова: 4-9.

## **ПОНЯТТЯ СОРТУВАННЯ. ВНУТРІШНЄ І ЗОВНІШНЄ СОРТУВАННЯ**

1. Основні визначення
2. Сортування методом прямого включення
3. Сортування методом прямого вибору
4. Сортування за допомогою прямого обміну (бульбашкове сортування)

Сортування - це розташування даних в пам'яті в регулярному вигляді за обраним параметром. Регулярність розглядають як зростання (спадання) значення параметра від початку до кінця масиву даних.

При обробці даних важливо знати інформаційне поле даних і розміщення їх в машині.

Розрізняють внутрішнє і зовнішнє сортування:

- внутрішнє сортування - сортування в оперативній пам'яті;
- зовнішнє сортування - сортування у зовнішній пам'яті.

Якщо відсортовані записи займають великий обсяг пам'яті, то їх переміщення вимагає великих витрат. Для того, щоб їх зменшити, сортування здійснюють в таблиці адрес ключів, тобто роблять перестановку покажчиків, а сам масив не рухається. Це - метод сортування таблиці адрес.

При сортуванні можуть зустрітися однакові ключі. У цьому випадку бажано після сортування розташувати однакові ключі в тому ж порядку, що і в вихідному файлі. Це - стійке сортування.

Ми будемо розглядати тільки сортування, яке не використовує додаткову оперативну пам'ять.

Ефективність сортування можна розглядати за кількома критеріями:

- час, що витрачається на сортування;
- обсяг оперативної пам'яті, необхідної для сортування;
- час, витрачений програмістом на написання програми.

Виділяємо перший критерій. Еквівалентом витраченого на сортування часу можна вважати кількість порівнянь і кількість переміщень при виконанні сортування.

- Порядок числа порівнянь і переміщень при сортуванні лежить в межах від  $O(n \log n)$  до  $O(n^2)$ ;

$O(n)$  - ідеальний і недосяжний випадок.

Розрізняють такі методи сортування:

- строгі (прямі) методи;
- поліпшені методи.

Строгі методи:

- метод прямого включення;
- метод прямого вибору;
- метод прямого обміну.

Ефективність цих методів приблизно однакова.

### **Рекомендована література.**

Базова: 1, 2.

Додаткова: 4-9.

### **Список рекомендованої літератури до дисципліни.**

#### **Базова**

1. Алгоритми і структура даних [Текст] : посібник / Н. Б. Шаховська, Р. О. Голощук. - Львів : Магнолія 2006, 2014. - 215 с. - (Серія "Комп'ютинг").
2. Вирт Никлаус. Алгоритмы и структуры данных : пер. с англ. / Н. Вирт. — 2-е изд., испр. — СПб. : Невский Диалект, 2001. — 351 с.
3. Математична логіка. Основи теорії алгоритмів [Текст] : навчальний посібник / С. С. Шкільняк. - К. : ДП "Видавничий дім "Персонал", 2009. - 280 с.

#### **Допоміжна**

4. Алгоритмы: вводный курс / Томас Х. Кормен : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2014. — 208 с.: ил.
5. Ахо Альфред В., Хопкрофт Джон, Ульман Джеффри Д. Структуры данных и алгоритмы.: Пер. с англ.: Уч.пос. – М.: Издательский дом «Вильямс», 2003. – 384 с.
6. Козак Л.І. Основи програмування : навчальний посібник / Л. І. Козак, І. В. Костюк, С. П. Стасевич. - Львів : Новий Світ-2000, 2012.
7. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ. - М.: Издательский дом «Вильямс», 2005.
8. Красиков И.В. Алгоритмы. Просто как дважды два./ И.В.Красиков, Е.И. Красикова. – М.: Эксмо, 2007. – 256 с.
9. Х. Ричард, К.Лоуренс. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста. Пер. с англ./ Х. Ричард, К.Лоуренс. – К.: Издательство «ДиаСофт», 2001.-736 с.

## ЗАВДАННЯ

### *до лабораторних робіт та організації самостійної роботи з дисципліни «Алгоритми і структури даних»*

#### **Лабораторна робота №1 Розробка лінійних алгоритмів**

**Мета роботи:** Дослідити та оволодіти навичками складання блок-схем, які використовують лінійні алгоритми.

Розрізняють три типові структури алгоритмів:

- лінійну;
- розгалужену;
- циклічну.

Найчастіше алгоритми вирішуваних завдань складається з окремих частин, які, в свою чергу, відносяться до однієї або іншої типовій структури.

У лінійній структурі символи алгоритму зображені на схемі в тій послідовності, в якій повинні бути виконані передбачені ними діями.

Всі виконані блок-схеми повинні відповідати станартам на оформлення схем і алгоритмів програм.

#### **Теоретичні питання до лабораторної роботи № 1:**

1. Поняття алгоритму.
2. Властивості алгоритму.
3. Етапи розробки і аналізу алгоритмів.
4. Основи аналізу ефективності алгоритмів.
5. Класи алгоритмів.
6. Способи подання алгоритмів.
7. Шаблони для подання алгоритму в графічному вигляді.

#### **Лабораторна робота №2 Розробка розгалужених алгоритмів**

**Мета роботи:** Придбання навичок побудови алгоритмів розгалуженої структури.

Розгалужена структура використовується в тому випадку, коли в залежності від значень проміжного результату вибирається та чи інша послідовність дій. Тобто в залежності від виконання деякої логічної умови обчислення повинно йти по одній або іншій гілці програми. На блок-схемі умова записується всередині ромба, з якого виходять дві стрілки, при виконанні умови шлях до наступної команди вказує лінія з написом «Так»; якщо умова не виконується, то шлях до наступної команди вказує стрілка з написом «Ні».

*Увага!!! Перед обчисленням значення математичного виразу, цей вислів слід проаналізувати: чи для всіх значень змінних його можна обчислити. В алгоритмі*

необхідно передбачити попередню перевірку змінних на значення, для яких вираз не може бути визначено. Якщо, наприклад, потрібно обчислити корінь парного ступеня, то потрібно перед обчисленням перевірити підкореневий вираз - він не повинний приймати негативні значення, а у випадку з дробом - перевірити знаменник на 0. У блок-схемі такі перевірки реалізуються за допомогою умовного блоку. Відсутність таких перевірок в програмі може призвести до критичних помилок.

#### Завдання.

1. Скласти алгоритм вирішення бікватратних рівняння  $ax^4+bx^2+c=0$  при заданих коефіцієнтах  $a$ ,  $b$  і  $c$ . Для вирішення бікватратного рівняння необхідно заміною  $y = x^2$  привести його до квадратного і вирішити це рівняння.
2. Скласти блок-схеми для вирішення заданих систем відповідно до варіанту. Побудувати графіки заданих функцій або поверхонь в MS Excel.

#### Теоретичні питання до лабораторної роботи №2:

1. Розгалужені алгоритми. Блок умови.
2. Поняття програми. Етапи, необхідні для вирішення завдань на ЕОМ.
3. Основні види процесів алгоритмізації.
4. Похибки алгоритмів.
5. Мови програмування.

### Лабораторна робота №3 Розробка циклічних алгоритмів

**Мета роботи:** Придбання навичок побудови і використання алгоритмів циклічної структури для вирішення задач.

Циклічна (ітеративна) структура - використовується, якщо при вирішенні задачі необхідно багаторазово проводити будь-які дії з різними вхідними даними, тобто передбачено неодноразове виконання однієї і тієї ж послідовності дій (тіла циклу). Число повторень в циклічних процесах або відомо заздалегідь, або залежить від будь-якої умови.

При організації циклів необхідно:

- 1) задати початкове значення параметра циклу - змінної, яка буде змінюватися при кожному повторенні циклу;
- 2) змінити значення параметра циклу при кожному повторенні циклу на крок зміни;
- 3) перевірити умову закінчення цих повторень за значенням цього параметра і перейти в початок циклу, якщо повторення ще не закінчені.

Управління циклом складається з трьох кроків: ініціалізації, перевірки і модифікації. При перевірці відстежується умова, яке позначає закінчення циклу, і якщо вона дає істину, виконання циклу завершується. Інші два етапи забезпечують прояв умови завершення. В процесі ініціалізації вводиться початкова умова, а крок модифікації видозмінює цю умову до умови завершення.

Зазвичай при написанні програм найбільше помилок допускається в управлінні циклом.

У циклі з передумовою (while, «поки») - перевірка умови завершення здійснюється до виконання тіла циклу. У циклі з умовою пост (repeat, «до») - до перевірки умови завершення виконується тіло циклу.

### Завдання.

1. Обчислити значення  $z$ , які відповідають кожному значенню  $X$  ( $x=1$ ;  $dx=0,5$ ) за формулою  $z = \ln \sqrt{\frac{x}{x^3 + 1}}$  (або іншою, наданою викладачем). Рахувати  $z$  до тих пір (і виводити значення  $x$ ,  $z$ ), поки підкореневий вираз більше або дорівнює  $b$  ( $b$  - вхідне дане). Визначити  $k$  - кількість обчислених  $z$ .
2. Обчислити значення  $g$ , відповідне кожному значенню  $x$  ( $x_n \leq x \leq x_k$ ,  $dx$ ) за формулою (або іншою, наданою викладачем):

$$g = \begin{cases} e^{-x} \sin(x), & |x| \leq a \\ e^{-x^2} \cos(x), & |x| \geq a \end{cases}$$

Знайти максимальне і мінімальне значення  $g$ .

### Теоретичні питання до лабораторної роботи №3:

1. Циклічні структури. Види циклів.
2. Поняття структур даних.
3. Класифікація структур.
4. Статичні і напівстатичні структури даних.

### Лабораторна робота №4 Алгоритми пошуку

**Мета роботи:** Навчитися, використовуючи алгоритми пошуку, знаходити корені рівнянь.

Алгоритм послідовного пошуку послідовно розглядає елементи списку в тому порядку, в якому вони розташовані в списку. Стоїть завдання знайти в списку задане значення. Алгоритм ефективний, коли мають справу з короткими списками. Якщо список впорядкований, то пошук продовжують до тих пір, поки в списку є імена і ім'я, що шукається, більше, ніж те, що розглядається на даному етапі ім'я списку. Якщо список не впорядкований, переглядають весь список. Загальна кількість порівнянь пропорційна кількості елементів в списку  $N$ .

Алгоритм двійкового пошуку. Якщо масив впорядкований, то тоді можна використовувати алгоритм бінарного (двійкового) пошуку. За результатом порівняння значення, що шукається, з середнім елементом масиву з розгляду виключають першу або другу половину списку, в залежності від того, більше або менше «середній» (що знаходиться в середині списку) елемент того елементу, який шукається. Отриману частину списку теж розбивають на дві частини і процедура повторюється. Таким чином, метод полягає в послідовному розподілі



розглянутого списку на сегменти до тих пір, поки не буде знайдений елемент або поле пошуку не звузиться до порожнього сегменту. Загальна кількість порівнянь пропорційно  $\log_2 N$ .

Алгоритм двійкового пошуку реалізується за допомогою рекурсивних структур. Цикл означає багаторазове виконання набору команд. Рекурсія ж передбачає повторне виконання набору команд як підзадачу самої себе.

#### Завдання.

Для рівняння  $x^3 - 2,92x^2 + 1,4355x + 0,791136 = 0$  (або іншого, наданого викладачем) виконати наступні завдання:

- а) побудувати поліном і визначити, на яких інтервалах знаходяться корені рівняння;
- б) знайти всі корені рівняння з використанням пошуку рішень Excel;
- в) вибрати один з сегментів і скласти блок-схему для знаходження кореня рівняння методом поділу відрізка навпіл і, використовуючи даний метод, знайти корінь в MS Excel.

#### Теоретичні питання до лабораторної роботи №4:

1. Поняття пошуку. Первинний і вторинний ключі. Пошук по заданому аргументу.
2. Послідовний пошук.
3. Ефективність послідовного пошуку.
4. Індексного-послідовний пошук і його ефективність.
5. Методи оптимізації пошуку.
6. Дерева бінарного пошуку.
7. Бінарний пошук (метод поділу навпіл).
8. Пошук за бінарним деревом зі вставкою.
9. Пошук за бінарним деревом зі включенням (вставкою).
10. Пошук за деревом з видаленням.

### Лабораторна робота №5 Алгоритми обробки масивів

**Мета роботи:** Ознайомитись з поняттям масиву. Набути навичок побудови алгоритмів обробки одновимірних масивів і матриць.

Обчислення суми елементів масиву.

Дано масив  $X$ , що складається з  $n$  елементів. Знайти суму елементів цього масиву. Процес накопичення суми елементів масиву досить простий і практично нічим не відрізняється від підсумовування значень деякої числової послідовності. Змінній  $S$  присвоюється значення рівне нулю, потім послідовно підсумовуються елементи масиву  $X$ . Блок-схема алгоритму розрахунку суми наведена на рис. 5.1.

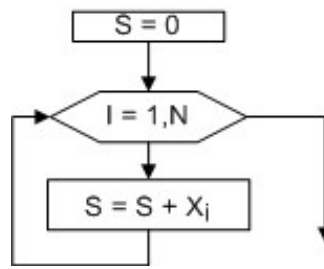


Рис. 5.1. Алгоритм обчислення суми елементів масиву

#### Обчислення добутку елементів масиву

Дано масив  $X$ , що складається з  $n$  елементів. Знайти добуток елементів цього масиву. Вирішення цього завдання зводиться до того, що значення змінної  $P$ , в яку попередньо була записана одиниця, послідовно множиться на значення  $i$ -го елемента масиву. Блок-схема алгоритму наведена на рис. 5.2.

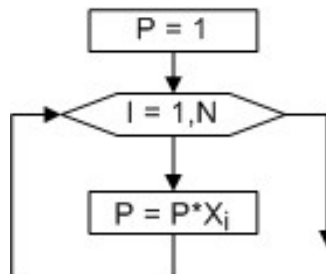


Рис. 5.2. Обчислення добутку елементів масиву

#### Завдання.

Заданий масив  $y$  з  $n$  цілих чисел. Сформувати масив  $z$  таким чином, щоб на початку йшли негативні елементи масиву  $y$ , потім позитивні і, нарешті, нульові (або в іншій послідовності, наданій викладачем).

#### Теоретичні питання до лабораторної роботи №5:

1. Поняття одномірних і двомірних масивів.
2. Обчислення суми і добутку елементів масиву.
3. Поняття головної та побічної діагоналей матриці.
4. Динамічні структури даних.

### Лабораторна робота №6 Алгоритми сортування масивів.

**Мета роботи:** Навчитися використовувати методи сортування з різними умовами для впорядкування структур даних.

Сортування є процес упорядкування елементів в масиві в порядку зростання або зменшення їх значень. Наприклад, масив  $X$  з  $n$  елементів буде впорядкований в порядку зростання значень його елементів, якщо  $X_1 \leq X_2 \leq \dots \leq X_n$ , і в порядку зменшення, якщо  $X_1 \geq X_2 \geq \dots \geq X_n$ .



Існує велика кількість алгоритмів сортування, але всі вони базуються на трьох основних:

- сортування обміном;
- сортування вибором;
- сортування вставкою.

#### Завдання.

1. Відсортувати масив за зростанням методом вибору.
2. Відсортувати масив за зростанням шляхом включення.
3. Відсортувати масив за зростанням методом обміну.
4. Відсортувати масив за зменшенням методом вибору.
5. Відсортувати масив за зменшенням шляхом включення.
6. Відсортувати масив за зменшенням методом обміну.
7. Видалити найбільший елемент масиву і впорядкувати його за зростанням.
8. Видалити найменший елемент масиву і впорядкувати його за зростанням.
9. Видалити найбільший елемент масиву і впорядкувати його за зменшенням.
10. Видалити найменший елемент масиву і впорядкувати його за зменшенням.

#### Теоретичні питання до лабораторної роботи №6:

1. Поняття сортування. Внутрішнє та зовнішнє сортування. Методи сортування.
2. Сортування методом прямого включення.
3. Сортування методом прямого вибору.
4. Сортування за допомогою прямого обміну (бульбашкове сортування).

#### Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1.	Розробка лінійних алгоритмів.	2
2.	Розробка розгалужених алгоритмів.	2
3.	Розробка циклічних алгоритмів.	2
4.	Алгоритми пошуку.	2
5.	Алгоритми обробки масивів.	2
6.	Алгоритми сортування масивів.	2
7.	Модульна контрольна робота	2

#### Самостійна робота студентів.

Розглянути і законспектувати відповіді на наступні питання:

1. Хеш-таблиці, хеш-функції і алгоритми хешування.
2. Рекурсія та її використання.
3. Алгоритми проходження дерев вглиб і вшир.
4. Види дерев (кореневі, збалансовані або АВЛ-дерева, червоно-чорні дерева), їх властивості та основні операції.

5. Алгоритми проходження графа (вглиб, вшир, найкоротша відстань між вершинами).

6. Бінарні купи. Пірамідальне сортування.

7. Алгоритми пошуку Ахо-Корасіка; Кнута, Моріса і Пратта; Рабина-Карпа; Боуера і Мура; Хорспула.

8. Поняття жадібного алгоритму. Відмінність між динамічним програмуванням і жадібним алгоритмом. Алгоритми: Краскала, Шенона-Фано, Хафмана, Прима.

### Розподіл балів, які отримують студенти

Модуль 1							МКР	Модуль2 (проект)	Сума
поточне тестування і самостійна робота									
ЗМ1	ЗМ2			ЗМ3		ЗМ1-3			
T1	T1	T2	T3	T1	T2	CPT			
8	9	9	10	12	10	15	12	15	100

CPT – теоретична самостійна робота

МКР – модульна контрольна робота

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	<b>A</b>	відмінно	зараховано
82-89	<b>B</b>	добре	
74-81	<b>C</b>		
64-73	<b>D</b>	задовільно	
60-63	<b>E</b>		
35-59	<b>FX</b>	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	<b>F</b>	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

### Питання до самоконтролю знань студентів.

1. Визначення інформації, алгоритму. Виконавці алгоритмів.
2. Способи опису алгоритмів. Властивості алгоритмів.
3. Поняття обчислювальної складності. Класи алгоритмів.

4. Лінійні, розгалужені і циклічні алгоритми.
5. Поняття структури даних. Рівні опису даних. Класифікація структур даних в програмах користувача і в пам'яті комп'ютера.
6. Основні види складених типів даних. Структури даних в пам'яті комп'ютера.
7. Поняття структури даних (СД) типу масив, набір допустимих операцій. Дескриптор СД типу масив. Ефективність масивів.
8. СД типу множина, запис (прямий декартовий добуток), таблиця, стек, черга, дек.
9. СД типу покажчик. Статичні і динамічні змінні.
10. СД типу лінійний однозв'язний список, циклічний лінійний список, двозв'язний лінійний список, багатозв'язний список.
11. Поняття хеш-функції. Алгоритми хешування.
12. Динамічне хешування.
13. Методи вирішення колізій. Переповнення таблиці і рехешування. Оцінка якості хеш-функції.
14. Визначення дерева. Бінарне дерево. Подання дерев у пам'яті комп'ютера. Алгоритми проходження дерев вглиб і вшир. Подання дерев у вигляді бінарних. Види бінарних дерев.
15. Поняття графа. Подання графа у пам'яті комп'ютера. Алгоритми проходження графа. Топологічне сортування. Пошук мостів. Найкоротша відстань між вершинами.
16. Загальна класифікація алгоритмів пошуку.
17. Лінійний пошук.
18. Бінарний пошук елементу масива.
19. Пошук методом Фібоначчі.
20. М-блоковий пошук.
21. Пошук в таблиці.
22. Прямий пошук рядка.
23. Алгоритми Ахо-Корасіка, Моріса-Прата, Кнута, Рабіна-Карпа, Боуера і Мура, Хорспула. Порівняння методів пошуку.
24. Сортування вставкою, вибором, поділом (Хоара), злиттям.
25. Метод Шелла.
26. Обмінне сортування.
27. Сортування за допомогою дерева.
28. Пірамідальне сортування.
29. Методи зовнішнього сортування: пряме злиття, природне злиття, збалансоване злиття за безліччю шляхів, багатофазне сортування.
30. Поняття і приклади жадібних алгоритмів.

## ЛІТЕРАТУРА

### Базова

1. Алгоритми і структура даних [Текст] : посібник / Н. Б. Шаховська, Р. О. Голошук. - Львів : Магнолія 2006, 2014. - 215 с. - (Серія "Комп'ютинг").

2. Вирт Никлаус. Алгоритмы и структуры данных : пер. с англ. / Н. Вирт. — 2-е изд., испр. — СПб. : Невский Диалект, 2001. — 351 с.
3. Математична логіка. Основи теорії алгоритмів [Текст] : навчальний посібник / С. С. Шкільняк. - К. : ДП "Видавничий дім "Персонал", 2009. - 280 с.

### **Допоміжна**

4. Алгоритмы: вводный курс / Томас Х. Кормен : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2014. — 208 с.: ил.
5. Ахо Альфред В., Хопкрофт Джон, Ульман Джеффри Д. Структуры данных и алгоритмы.: Пер. с англ.: Уч.пос. – М.: Издательский дом «Вильямс», 2003. – 384 с.
6. Вирт, Никлаус. Алгоритмы и структуры данных : пер. с англ. / Н. Вирт. — 2-е изд., испр. — СПб. : Невский Диалект, 2001. — 351 с.
7. Красиков И.В. Алгоритмы. Просто как дважды два./ И.В.Красиков, Е.И. Красикова. – М.: Эксмо, 2007. – 256 с.
8. Х. Ричард, К.Лоуренс. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста. Пер. с англ./ Х. Ричард, К.Лоуренс. – К.: Издательство «ДиаСофт», 2001.-736 с.

### **Інформаційні ресурси**

9. Алгоритмы и структуры данных. Лекториум  
<https://www.lektorium.tv/lecture/13343>
- 10.Алгоритмы и структуры данных. Видеолекции Технопарка.  
<http://habrahabr.ru/company/abbyy/blog/251561/>
- 11.Типичный программист. Алгоритмы и структуры данных для начинающих.  
<http://tproger.ru/tag/for-beginners/>

**Засоби діагностики навчальних досягнень студентів  
з дисципліни «Алгоритми та структури даних»**

**Теоретичні питання до лабораторної роботи № 1:**

8. Поняття алгоритму.
9. Властивості алгоритму.
10. Етапи розробки і аналізу алгоритмів.
11. Основи аналізу ефективності алгоритмів.
12. Класи алгоритмів.
13. Способи подання алгоритмів.
14. Шаблони для подання алгоритму в графічному вигляді.

**Теоретичні питання до лабораторної роботи №2:**

6. Розгалужені алгоритми. Блок умови.
7. Поняття програми. Етапи, необхідні для вирішення завдань на ЕОМ.
8. Основні види процесів алгоритмізації.
9. Похибки алгоритмів.
10. Мови програмування.

**Теоретичні питання до лабораторної роботи №3:**

1. Циклічні структури. Види циклів.
2. Поняття структур даних.
3. Класифікація структур.
4. Статичні і полустатичні структури даних.

**Теоретичні питання до лабораторної роботи №4:**

11. Поняття пошуку. Первинний і вторинний ключі. Пошук по заданому аргументу.
12. Послідовний пошук.
13. Ефективність послідовного пошуку.
14. Індексного-послідовний пошук і його ефективність.
15. Методи оптимізації пошуку.
16. Дерева бінарного пошуку.
17. Бінарний пошук (метод поділу навпіл).
18. Пошук за бінарним деревом зі вставкою.
19. Пошук за бінарним деревом зі включенням (вставкою).
20. Пошук за деревом з видаленням.

**Теоретичні питання до лабораторної роботи №5:**

5. Поняття одномірних і двомірних масивів.
6. Обчислення суми і добутку елементів масиву.
7. Поняття головної та побічної діагоналей матриці.
8. Динамічні структури даних.

### **Теоретичні питання до лабораторної роботи №6:**

1. Поняття сортування. Внутрішнє та зовнішнє сортування. Методи сортування.
2. Сортування методом прямого включення.
3. Сортування методом прямого вибору.
4. Сортування за допомогою прямого обміну (бульбашкове сортування).

### **Питання до самоконтролю знань студентів.**

1. Визначення інформації, алгоритму. Виконавці алгоритмів.
2. Способи опису алгоритмів. Властивості алгоритмів.
3. Поняття обчислювальної складності. Класи алгоритмів.
4. Лінійні, розгалужені і циклічні алгоритми.
5. Поняття структури даних. Рівні опису даних. Класифікація структур даних в програмах користувача і в пам'яті комп'ютера.
6. Основні види складених типів даних. Структури даних в пам'яті комп'ютера.
7. Поняття структури даних (СД) типу масив, набір допустимих операцій. Дескриптор СД типу масив. Ефективність масивів.
8. СД типу множина, запис (прямий декартовий добуток), таблиця, стек, черга, дек.
9. СД типу покажчик. Статичні і динамічні змінні.
10. СД типу лінійний однозв'язний список, циклічний лінійний список, двозв'язний лінійний список, багатозв'язний список.
11. Поняття хеш-функції. Алгоритми хешування.
12. Динамічне хешування.
13. Методи вирішення колізій. Переповнення таблиці і рехешування. Оцінка якості хеш-функції.
14. Визначення дерева. Бінарне дерево. Подання дерев у пам'яті комп'ютера. Алгоритми проходження дерев вглиб і вшир. Подання дерев у вигляді бінарних. Види бінарних дерев.
15. Поняття графа. Подання графа у пам'яті комп'ютера. Алгоритми проходження графа. Топологічне сортування. Пошук мостів. Найкоротша відстань між вершинами.
16. Загальна класифікація алгоритмів пошуку.
17. Лінійний пошук.
18. Бінарний пошук елементу масива.
19. Пошук методом Фібоначчі.
20. М-блоковий пошук.
21. Пошук в таблиці.
22. Прямий пошук рядка.
23. Алгоритми Ахо-Корасіка, Моріса-Прата, Кнута, Рабіна-Карпа, Боуера і Мура, Хорспула. Порівняння методів пошуку.
24. Сортування вставкою, вибором, поділом (Хоара), злиттям.
25. Метод Шелла.
26. Обмінне сортування.
27. Сортування за допомогою дерева.

- 28.Пірамідальне сортування.
- 29.Методи зовнішнього сортування: пряме злиття, природне злиття, збалансоване злиття за безліччю шляхів, багатофазне сортування.
- 30.Поняття і приклади жадібних алгоритмів.

**Інформаційні матеріали бібліотеки по забезпеченню навчальними підручниками (посібниками) з дисципліни**

з/п	Найменування навчальної дисципліни	Автор підручника (навчального посібника тощо)	Найменування підручника (навчального посібника тощо)	Найменування видавництва, рік видання	Кількість примірників
1	2	3	4	5	6
1	Алгоритми та структури даних	Вирт Н.	Алгоритмы и структуры данных : Пер. с англ.-2-е изд., испр.	СПб: Невский Диалект, 2001.	21
		Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К.	Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ.	М.: Издательский дом «Вильямс», 2005.	10
		Козак Л.І.	Основи програмування : навчальний посібник / Л. І. Козак, І. В. Костюк, С. П. Стасевич.	Львів : Новий Світ-2000, 2012.	22
		Н. Б. Шаховська, Р. О. Голощук.	Алгоритми і структура даних : посібник	Львів : Магнолія 2006, 2014. - 215 с. - (Комп'ютинг)	15
		Шкільняк С.С.	Математична логіка. Основи теорії алгоритмів : навчальний посібник	"Видавничий дім "Персонал", 2009.	12