

Алгоритми та структури даних в підготовці інженерів-програмістів

Сьогодні, кажучи про майбутній розвиток ІТ-технологій і комп'ютеризації після 2010 року, можна відзначити, що світ стоїть перед новими відкриттями: у наше життя входять хмарні технології, людина тепер може управляти комп'ютером мімікою, голосом, рухами тіла, через п'ять років будь-яка поверхня зможе бути використана як дисплей, можливо, з'явиться квантовий комп'ютер.

У зв'язку з цим, з одного боку, у навчанні активно почнуть використовуватися всі ІТ-нововведення, з іншого – з'явиться необхідність підготовки фахівців нового покоління, які володіють необхідними фундаментальними знаннями для успішного продовження розвитку ІТ-технологій. Разом з тим незважаючи на неймовірно високу динаміку змін в ІТ-сфері, формування необхідних професійних якостей у студентів комп'ютерних спеціальностей, як і раніше буде залежати від вивчення базисного корпусу знань ВОК (Body of Knowledge) [1, 31-40], що становить фундаментальне ядро комп'ютерних спеціальностей. ВОК включає чотирнадцять базисних тематик, серед яких є тематика АЛ «Алгоритми і складність» [1, 31-47].

Серед педагогічних проблем навчання важливих питань є правильний і педагогічно виважений добір і структурування навчального матеріалу. Для створення програмних продуктів нового покоління будуть потрібні розробники, які володіють знаннями простих і складних алгоритмів, простих і складних структур даних, методик програмування, проектування і розробки. Тому в підготовці інженерів-програмістів слід приділяти особливу увагу цим питанням.

З причини інтенсивного розвитку комп'ютерної науки зміст дисциплін фундаментального ядра підготовки фахівців комп'ютерних спеціальностей, а також методичні системи навчання відповідних дисциплін є актуальним предметом дослідження.

Зміст деяких дисциплін комп'ютерної спрямованості та методичні системи навчання цих дисциплін є добре вивченим питанням. Наприклад, останні два десятиліття досить активно розглядалася і розроблялася методична система навчання інформатики як навчальної дисципліни.

У роботі [2] розглянуто питання, пов'язані з необхідністю вивчення принципів об'єктно-орієнтованого програмування (ООП) студентами комп'ютерних спеціальностей за допомогою різних мов програмування, наприклад, Java і C++. Крім того, в роботі [2] обговорюються особливості реалізації об'єктно-орієнтованого підходу з використанням об'єктно-орієнтованих мов Smalltalk, Eiffel, C++, Java, а також мов проектування UML та ОО-технології CORBA. Ентоні Елієнс в роботі [2] пропонує наступну формулу ОО-підходу:

$$\text{ОО-підхід} = \text{Об'єкти} + \text{Успадкування} + \text{Пересилання повідомлення}$$

Девід Вест в книзі [3] описує об'єктне мислення, необхідне майбутньому розробнику програмного забезпечення, і яке, як вважається, притаманне розробникам з компанії Microsoft. Девід Вест вважає, що саме об'єктне мислення, знання алгоритмів і структур даних робить програміста хорошим професіоналом, а не методи та інструменти. На прикладах він показує, що кращі програмісти спираються на аналіз і концептуальність в мисленні більше, ніж на формальні процеси та методи.

У класичній монографії Дональда Кнута «Искусство программирования» [4], [5], [6] описані основні поняття і методи програмування, наведені найважливіші алгоритми, в тому числі напівчисельні, а також структури даних комп'ютеризації.

Однак аналіз педагогічних праць і літератури виявив маловивчені питання добору та структурування змісту дисциплін базисного корпусу знань за напрямом підготовки «Інформатика», враховуючи швидкість змін, що відбуваються у світі комп'ютеризації, і необхідність їх адекватного відображення в навчальному процесі.

В зв'язку з цим важливо описати підходи до навчання алгоритмів і структур даних, що дозволяє формувати у студентів комп'ютерних спеціальностей об'єктний стиль мислення, а також дати рекомендації до структурування змісту дисциплін «Алгоритми та структури даних» та «Удосконалені методи розробки та аналізу», що вивчаються студентами комп'ютерних спеціальностей.

Розробка та побудова алгоритму, знаходження ефективного алгоритму, вибір відповідного

способу зберігання і організації даних (структури даних) – це фундаментальні питання комп'ютерної науки. Багато підходів і методів аналізу та побудови різних алгоритмів розроблялися ще на зорі комп'ютерної ери і, не зважаючи на це, залишаються актуальними незалежно від розвитку комп'ютерних технологій. Тому дисципліна «Алгоритми та структури даних» є однією з найважливіших дисциплін при підготовці майбутніх фахівців в галузі комп'ютерингу в Україні та за кордоном. Навчання алгоритмів і структур даних, а також вдосконалених алгоритмів, методів розробки та аналізу алгоритмів має надзвичайну практичну значимість при підготовці студентів напрямку підготовки 6.040302 «Інформатика».

Вивченню сучасних алгоритмів, розробці програм, аналізу алгоритмів, структур даних присвячена класична книга авторів Томаса Кормена, Чарльза Лейзерсона, Рональда Ривеста, Кліффорда Штайна «Алгоритмы: построение и анализ» [7], [8], яка зазвичай рекомендується в якості основного підручника з дисципліни «Алгоритми та структури даних». Перше видання цієї книги [9], авторами якої були Т. Кормен, Ч. Лейзерсон, Р. Ривест, було опубліковано в 1990 році. Серед американських студентів і викладачів ця книга називалася «Big White Book» (велика біла книга), оскільки вона була білого кольору. Також це видання називалася «CLR» (за першими літерами прізвищ авторів; читається «сі-ел-ар»). Переклад цього видання на російську мову з'явився в СНД у 2000 році. Друге видання книги «Алгоритмы: построение и анализ» [10] було опубліковано видавництвом Массачусетського технологічного інституту (MIT) в 2001 році, і серед студентів і викладачів називалося «CLRS» за першими літерами прізвищ авторів. У цьому виданні додався ще один автор – Кліффорд Штайн, було внесено багато змін, були додані нові розділи, вправи та завдання. Російський переклад цього видання [8] з'явився в 2005 році. У 2009 році підготовлено третє видання цієї книги [11]. У третьому виданні [11] автори додали в якості прикладу, що ілюструє парадигму «розділяй і володарюй», алгоритм Штрассе для множення матриць. Також додані нові алгоритми, наприклад, алгоритми, пов'язані з деревами ван Емд Боаса (van Emde Boas tree), які набагато ефективніше збалансованих двійкових дерев пошуку для задач пошуку елемента, що належить заданому інтервалу цілих чисел. Древа ван Емд Боаса вперше були запропоновані Пітером ван Емд Боас в 1977 році, опубліковані в статтях [12], [13], але раніше в навчальний матеріал дисципліни «Алгоритми та структури даних», як правило, не включалися.

У роботі [14] нами була зроблена спроба описати підходи до навчання дисципліни «Алгоритми та структури даних», що входить у нормативну частину циклу професійно-орієнтованих дисциплін напряму підготовки 6.040302 «Інформатика». Були наведені різні навчальні компонування і реалізації.

Проектування змісту навчальної дисципліни «Алгоритми та структури даних» дозволило встановити число і зміст модулів; визначити співвідношення теоретичних і лабораторних занять в кожному з модулів; зміст самостійної роботи студентів напряму підготовки «Інформатика». Дисципліна «Алгоритми та структури даних» містить 5 кредитів ECTS, вивчається у п'ятому семестрі і включає в себе три модулі: «Математичний апарат аналізу алгоритмів», «Алгоритми сортування та завдання вибору», «Структури даних». Для успішного вивчення цієї дисципліни студенти повинні попередньо пройти навчальні курси з дисциплін «Дискретна математика», «Програмування», «Теорія ймовірностей і математична статистика».

Головним завданням навчання дисципліни «Алгоритми та структури даних» є розгляд методів аналізу і побудови класичних алгоритмів, використовуваних для розв'язування практичних задач і ознайомлення з основними структурами даних. Розв'язування поставленої задачі представляється можливим у результаті навчання дисципліни за схемою, поданою на рис. 1. Вивчення теоретичних основ здійснюється наступним послідовним розглядом трьох основних розділів – «Математичний апарат аналізу алгоритмів», «Алгоритми сортування та завдання вибору» і «Структури даних», кожен з яких оформлено як окремі теми.

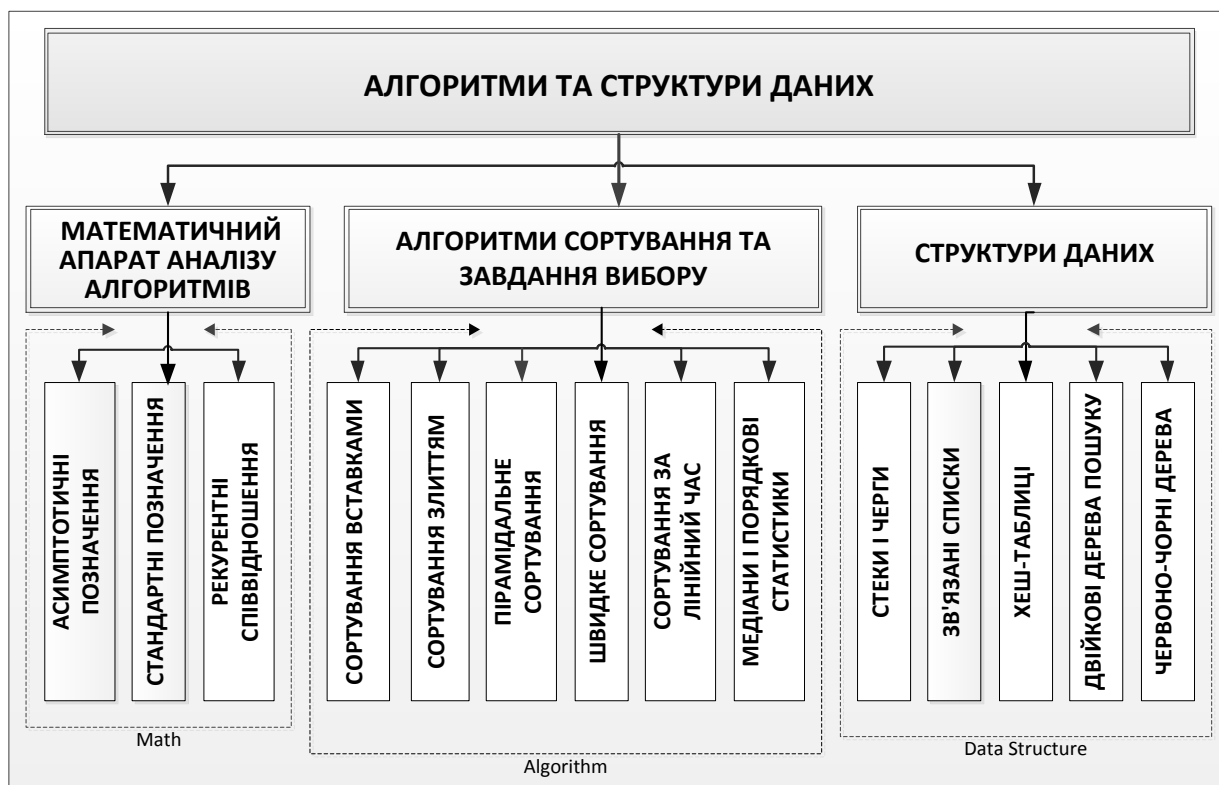


Рис.1. Схема вивчення дисципліни «Алгоритми та структури даних»

Вибір і побудова ефективного алгоритму – одне з основних професійних умінь інженера-програміста. Час роботи алгоритму і обсяг необхідних комп'ютерних ресурсів, наприклад, пам'яті, є характеристиками ефективності алгоритму, що використовується для розв'язування прикладної задачі. За допомогою досліджуваного у рамках дисципліни «Алгоритми та структури даних» математичного апарату можна проаналізувати часову ефективність алгоритму і дати оцінку ефективності використання комп'ютерних ресурсів. Математичний апарат аналізу алгоритмів включає в себе такі питання: найдовше і середній час роботи за алгоритмом, асимптотичні позначення (Θ -позначення, O -позначення, Ω -позначення), рекурентні співвідношення.

Використання сучасної комп'ютерної техніки дозволяє вирішувати практично будь-які проблеми завдяки потужним обчислювальним характеристикам, швидкодії і великим обсягам пам'яті. Незважаючи на це, ефективний алгоритм як інструмент розв'язування обчислювальної задачі, продовжує залишатися такою ж значущою технологією, як і сучасне апаратне забезпечення комп'ютера. Тому для вирішення все більш складних завдань необхідні знання ефективних алгоритмів та структур даних.

Вивчення класичних алгоритмів побудовано на розгляді задач сортування як найбільш «показових» обчислювальних задач, що дозволяє розглянути різні методи розробки та аналізу алгоритмів.

Дослідження в навчальній дисципліні алгоритмів сортування та структур даних дозволяє вирішити основні завдання курсу:

- виконати часове оцінювання алгоритму, використовуючи асимптотичні позначення і рекурентні співвідношення;
- навчитися вибирати найбільш відповідні для розв'язування обчислювальної задачі алгоритм і структури даних;
- проаналізувати ефективність використання комп'ютерних ресурсів, у тому числі пам'яті;
- вивчити основні методи, використовувані для розробки та аналізу алгоритмів в програмній інженерії.

Навички аналізу алгоритмів і методів програмування використовуються надалі в процесі вивчення дисциплін «Удосконалені методи розробки та аналізу», «Алгоритми комп'ютерної анімації», «Теорія програмування», «Технологія програмування», «Технологія проектування».

Дисципліна «Удосконалені методи розробки та аналізу» є продовженням дисципліни «Алгоритми та структури даних» і входить до варіативної частини циклу професійно-орієнтованих дисциплін підготовки за напрямом 6.040302 «Інформатика». Основне завдання її навчання –

оволодіння знаннями про розширені алгоритми та структури даних, що використовуються в сучасній програмній інженерії, а також закріплення знань і практичне застосування навичок, отриманих при вивченні дисципліни «Алгоритми та структури даних».

При цьому пропонується вивчати дисципліну «Удосконалені методи розробки та аналізу» відповідно до схеми, зображеної на рис. 2. Теоретична частина включає три модулі – «Методи розробки», «Складні структури даних» та «Робота з графами», кожен з яких поділений на окремі теми. У модулі «Методи розробки» розглядаються три важливих методи розробки та аналізу ефективних алгоритмів – динамічне програмування, жадібні алгоритми і амортизаційний аналіз.

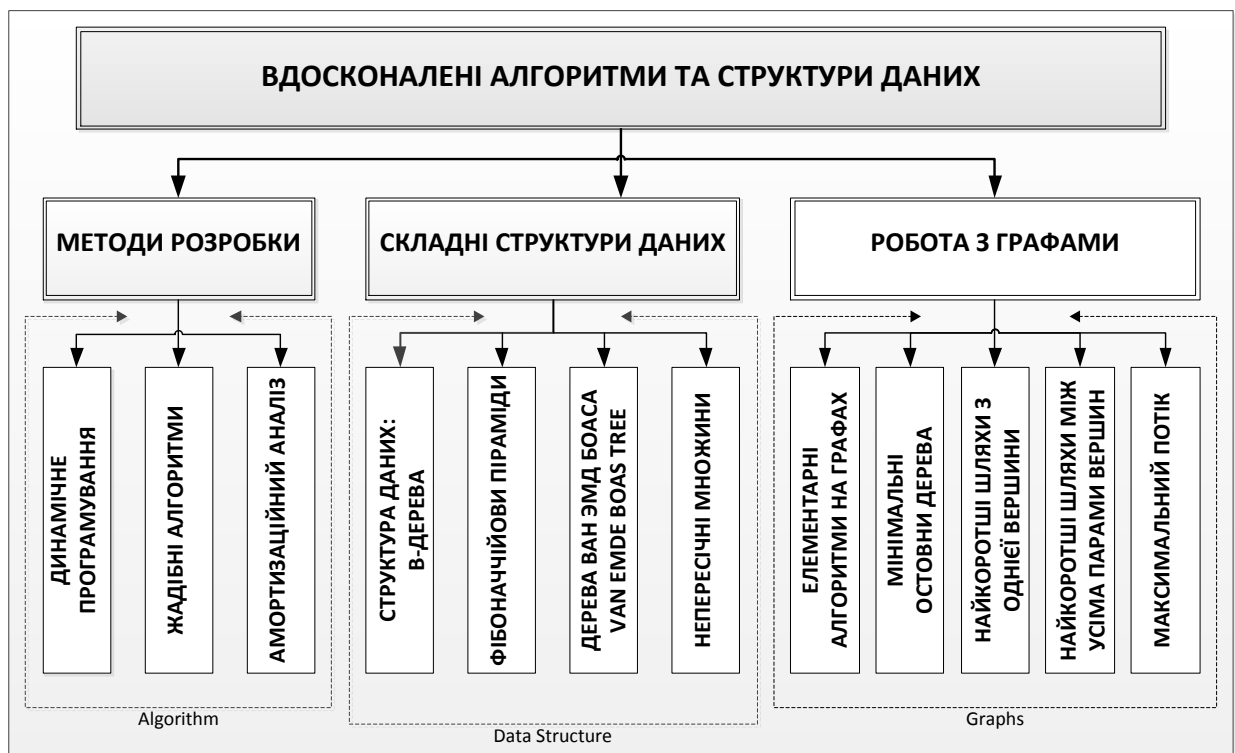


Рис. 2. Схема вивчення дисципліни «Удосконалені методи розробки та аналізу»

У модулі «Складні структури даних» розглядаються складні структури даних, що використовуються при здійсненні операцій над динамічними множинами – В-дерева, фібоначчійові піраміди, дерева ван Емд Босесі, непересічні множини.

Модуль «Робота з графами» включає в себе алгоритми для роботи з графами; розглядаються методи подання графа, алгоритми отримання відомостей про структуру шляхом обходу графа, пошук в ширину і в глибину, топологічне сортування.

Закріплення та удосконалення практичних навичок програмування, розробки та аналізу алгоритмів здійснюється в ході лабораторних робіт під час програмної реалізації досліджуваного алгоритму. Лабораторні роботи в ході вивчення дисципліни «Алгоритми та структури даних» проводяться за чіткою схемою навчання, основні етапи якої відображені на рис. 3.

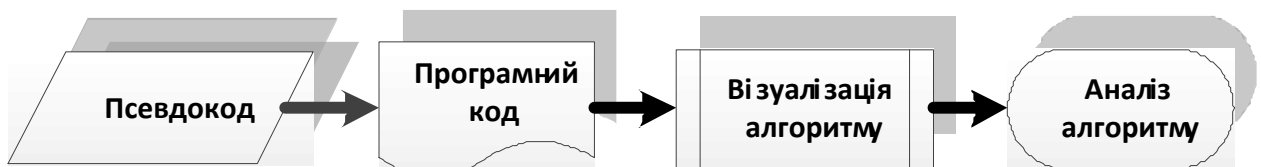


Рис. 3. Схема проведення лабораторних занять з дисципліни «Алгоритми та структури даних»

Розглянемо поетапно хід виконання лабораторної роботи з дисципліни «Алгоритми та структури даних». Для інженера-програміста сам алгоритм розв'язування задачі завжди важливіший, ніж його опис якою-небудь мовою програмування. Тому перш ніж перейти до написання програмного коду, необхідно описати формальною мовою алгоритм розв'язування задачі. Така мова опису не повинна бути прив'язана до конкретної мови програмування, але

повинна бути зрозумілою людині, знайомій з алгоритмізацією. В якості такої мови може бути використаний псевдокод, що дозволяє опустити технічні деталі (наприклад, введення послідовності, виведення на екран) і зосередитися на логіці відшукування розв'язку. Відсутність або недостатнє знання синтаксису мови програмування можна заповнити вивченням відповідних мовних конструкцій і правил програмування. Відсутність навичок створення робочого алгоритму не дозволить інженеру-програмісту написати ефективний програмний код, якою б мовою програмування він не володів.

У пропонованих завданнях лабораторних робіт входить опис алгоритму саме на псевдокоді, який потім реалізується з використанням будь-якої мови програмування.

При вивченні дисципліни «Удосконалені методи розробки та аналізу» на лабораторних заняттях студентам необхідно описати якою-небудь мовою програмування і подати викладачеві програмний код, відповідний досліджуваному алгоритму. На лабораторних заняттях з дисципліни «Алгоритми та структури даних» студентам пропонується програмувати кількома мовами програмування, наприклад, Pascal, C++, Java, Python, C#. Таке варіювання дозволяє студентам аналізувати не тільки алгоритми, але і звернути увагу на ефективність програмної реалізації, порівняти реалізації алгоритму, описаного різними мовами програмування.

Після програмної реалізації завдання та отримання програмного коду для більш глибокого розуміння алгоритму або структури даних студентів необхідно виконати їх візуалізацію. Будь-яка візуалізація – це перш за все засіб аналізу, особливо, якщо мова йде про абстрактні дані. Саме тому візуалізація розглядається як додатковий наочний засіб аналізу досліджуваних алгоритмів і структур даних. В залежності від способу відображення доцільно виконувати візуалізацію наступними способами:

- звичайне зображення – зображення або кілька зображень роботи алгоритму або структури даних у необхідному ракурсі;
- анімація – зображення об'єкта в динаміці за розробленим для демонстрації сценарієм;
- напівавтоматичне виконання – включає можливості анімації та мультимедіа, а також управління досліджуваним об'єктом (наприклад, вилучення, вставлення, пошук елемента).

Студентам звичайно ставляться при розробці нових анімацій вимоги дотримуватися наступних правил: естетичність анімації, наочність, простота розуміння анімації, можливість управляти об'єктами анімації.

Заключний етап виконання лабораторної роботи і власне вивчення алгоритму або структури даних – аналіз. Тут особливо важливо вміти використовувати математичний апарат, досліджуваний у рамках дисципліни «Алгоритми та структури даних». Студенту необхідно вміти знаходити мінімальний, максимальний, середній час виконання алгоритму, виконувати асимптотичні оцінки, аналізувати рекурентні співвідношення за методами підстановки, за допомогою основної теореми (Master Theorem) [11, 93-107].

Вивчення і практичне використання в процесі програмування фундаментальних алгоритмів і структур даних дозволяє сформувати необхідні професійні якості і навички у студентів комп'ютерних спеціальностей, а саме: об'єктне мислення, концептуальність в мисленні, об'єктний стиль програмування, а також володіння різними методиками програмування, проектування і розробки програмних продуктів.

Педагогічно виважений добір і структурування навчального матеріалу стосовно алгоритмів і структур даних можливий з урахуванням сучасних ІТ-тенденцій і потенціалу комп'ютерної науки та фундаментальних напрацювань у програмній інженерії.

Доцільною також є розробка графів квантів навчального матеріалу з дисциплін «Алгоритми та структури даних», «Удосконалені методи розробки та аналізу», реалізація алгоритмів і структур даних, описаних не однією, а кількома мовами програмування, такими як Python, Java, C++, C# з метою аналізу ефективності використання тієї чи іншої мови при розв'язуванні практичних задач програмної інженерії.

Література

1. Сейдаметова З.С. Подготовка инженеров-программистов по специальности «Информатика»: [монография] / Зарема Сейдалиевна Сейдаметова. – Симферополь: Крымчпедгиз, 2007. – 480, [1] с.
2. Элиенс А. Принципы объектно-ориентированной разработки программ. 2-е издание. – М.: Издательский дом «Вильямс», 2002. – 496 с.
3. West D. Microsoft Object Thinking – Redmond (Washington): Microsoft Press, 2004. – 368 p.

4. Кнут Д.Э. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ.: Уч. пос. / Дональд Эрвин Кнут. – М.: Издательский дом «Вильямс», 2000. – 720 с.
5. Кнут Д.Э. Искусство программирования, том 2. Получисленные алгоритмы, 3-е изд.: Пер. с англ.: Уч. пос. / Дональд Эрвин Кнут. – М.: Издательский дом «Вильямс», 2000. – 832 с.
6. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд.: Пер. с англ.: Уч. пос. / Дональд Эрвин Кнут. – М.: Издательский дом «Вильямс», 2000. – 832 с.
7. Кормен Т.Х. Алгоритмы: построение и анализ, 1-е издание.: Пер. с англ. / Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест. – М.: МЦНМО, 2000. – 960 с.
8. Кормен Т.Х. Алгоритмы: построение и анализ, 2-е издание.: Пер. с англ. / Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест, К. Штайн. – М.: Издательский дом «Вильямс», 2005. – 1296 с.
9. Cormen T.H. Introduction to Algorithms / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein; First Edition. – The MIT Press, 1990. – 1048 p.
10. Cormen T.H. Introduction to Algorithms / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein; Second Edition. – The MIT Press, 2001. – 1202 p.
11. Cormen T.H. Introduction to Algorithms / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein; Third Edition. – The MIT Press, 2009. – 1312 p.
12. Emde Boas P. Preserving order in a forest in less than logarithmic time and linear space / Peter van Emde Boas. // Information Processing Letters, # 6(3) 1977. – pp. 80-82.
13. Emde Boas P. Design and implementation of an efficient priority queue / P. van Emde Boas, R. Kaas, E. Zijlstra. // Math. Systems Theory, #10, 1977. – pp. 99-127.
14. Меджитова Л.М. Подходы к обучению алгоритмам и структурам данных / Л.М. Меджитова, З.С. Сейдаметова, С.Н. Сейтвелиева // Январские педагогические чтения: современное образование как процесс формирования человека, общества, государства. – Симферополь: КРП «Издательство «Крымучпедгиз», 2010. – С. 119-123.