

творчість, інноваційний менеджмент); використання інновацій у педагогічній практиці (інтеграція різнотипних і різнорівневих ресурсів для індивідуалізації навчання і комунікації для роботи в команді); наукового забезпечення педагогічних інновацій (цілі створення нових курсів, методи та технології навчання, апробація, впровадження) та освоєння технології педагогічного експерименту.

У різні роки фахівці кафедри інформатики та методики її навчання розробляли такі дисципліни за вибором: «Вивчення та використання відкритого програмного забезпечення», «Розробка Web-додатків навчального призначення», «Адміністрування навчальних систем», «Розробка електронних навчальних ресурсів: методика і технологія», «Сучасні технології дистанційного навчання», «Технологія розробки педагогічних комп'ютерних тестів», «Сучасні сервіси Інтернет у педагогічній практиці (Web 2.0)», «Інтеграція інформаційних та освітніх технологій XXI століття» тощо.

Висновки

1. У зв'язку із переходом на 1,5–2 річний термін навчання у професійній магістратурі актуальною є розробка нових стандартів вищої освіти для освітньо-кваліфікаційного рівня «магістр».

2. У теоретичній підготовці магістрів за спеціальністю «Інформатика» вибіркові дисципліни за кількістю кредитів (60% від обсягу теоретичного навчання) переважають нормативні.

3. У процесі формування змісту вибіркових дисциплін доцільно дотримуватись таких концепцій: концепції випереджаючої освіти, концепції фундаменталізації освіти, концепції інноватики в освіті.

4. Вибіркові дисципліни професійної магістратури зі спеціальності «Інформатика» з кваліфікацією «Викладач інформатики» доцільно формувати як короткі і динамічні за змістом спецкурси.

Список використаних джерел

1. Высшее образование в XXI веке: подходы и практические меры. Всемирная декларация о высшем образовании для XXI века. – М.: СГУ, 1999. – 36 с.

2. Ильин И.В., Урсул А.Д. Эволюционная глобалистика (концепция эволюции глобальных процессов). – М.: Изд-во МГУ, 2009. – 232 с.

3. Жалдак М.І. Деякі методичні аспекти навчання інформатики в школі і педагогічному університеті // Комп'ютерно-орієнтовані системи навчання. Випуск 9. Науковий часопис. – Київ.: НПУ ім. М.П. Драгоманова. 2005. – С. 3–14.

4. Кинелев В.Г. Образование и цивилизация [Электронный ресурс]. – Режим доступа: <http://www.emag.iis.ru/arc/infosoc/emag.nsf/BPA/> – Название с экрана.

5. Колин К. К. Социальная информатика: Учебное пособие для вузов. – М.: Академический Проект; М.: Фонд «Мир», 2003. – 432 с.

6. Морзе Н.В. Як навчати вчителів, щоб комп'ютерні технології перестали бути дивом у навчанні? / Н. В. Морзе // Комп'ютер у школі та сім'ї. – 2010. – №6 (86). – С. 10–14.

7. Про Концепцію організації підготовки магістрів в Україні. Наказ МОН № 99 від 10.02.10 року [Електронний ресурс]. – Режим доступу: http://osvita.ua/legislation/Vishya_osvita/6670/ – Назва з екрана.

8. Рамський Ю.С., Резіна О.В. Вивчення інформаційно-пошукових систем мережі Інтернет. Пошук-МЕТА: Посібник для вчителів. Навчально-програмний комплекс. – К.: НПУ імені М.П. Драгоманова, 2004. – 141 с.

Сейдаметова З.С.

Кримський інженерно-педагогічний університет

Складні структури даних в навчанні інженерів-програмістів

Постановка проблеми. Алгоритми і структури даних є фундаментом комп'ютерних наук та програмної інженерії. Будь-яка комп'ютерна система, апаратне забезпечення, система реального часу, система зберігання даних залежать від правильно обраних алгоритмів, структур даних, ефективності алгоритмічних реалізацій. Вивчення алгоритмів з одного боку дозволяє розглянути проблему зсередини, ретельно вивчивши її внутрішню природу, а з іншого боку дає можливість реалізовувати алгоритми з урахуванням особливостей мов програмування, апаратного забезпечення, парадигм програмування.

Як зазначено в статті [1], в двадцять першому столітті відбулися зміни в структурі робочих місць, вони перестали бути жорстко контрольованими, стали більш мобільними. У зв'язку з цим в двадцять першому столітті з'явилася необхідність у співробітниках, здатних до самоорганізації та самоуправління, які вмюють ставити завдання і їх вирішувати, продуктивно управляти своїм часом. Для підготовки фахівців, потрібних в двадцять першому столітті, освіта повинна сприяти, як зазначено в [1, с. 58-59] розвитку шести навичок. Це чотири навички групи «4Cs» – критичне

мислення, комунікація, співпраця, креативність (critical thinking, communication, collaboration, creativity), в комбінації з умінням «самоспрямування» і розумінням проблем у глобальній шкалі.

Підготовка інженерів-програмістів, здатних вирішувати проблеми двадцять першого століття, повинна спиратися не тільки на технічні та програмістські аспекти, але й на формування релевантних, сучасних знань, вмінь і навичок.

Аналіз досліджень і публікацій. Вивчення алгоритмів і структур даних в підготовці інженерів-програмістів є важливим аспектом професійної підготовки, цим питанням приділяють увагу багато дослідників. У статті [2], присвяченій інструментарію для вивчення вдосконалених структур даних, охарактеризована система навчання студентів різних реалізацій В-дерев. У статті [3] запропонований ігровий підхід до ознайомлення зі структурами даних, при цьому в якості інструмента використовуються елементи «спортивного» програмування.

У статті [4], присвяченій загальним проблемам вивчення дисципліни «Алгоритми і структури даних», розглянуто підхід до формування контенту, а також методик навчання. У роботах [5] – [17] охарактеризовані класичні складні структури даних, описані алгоритми реалізації основних операцій. У статті [18] наведено порівняння і оцінка двох складних структур даних – AVL та RBT дерев. У статті [19] описаний альтернативний підхід до зберігання та вилучення даних. У статтях [20], [21] охарактеризовані способи аналізу структур даних, реалізованих за відомими моделями, а також показані тестові кейси для перевірки правильності реалізації. У статтях [22], [23] виявлені ефективні алгоритми для опису операцій вилучення в збалансованих деревах пошуку без необхідності подальшого балансування. У статтях [24] – [27] описані задачі, в яких є необхідність використання реалізацій організації зберігання даних за допомогою складних структур даних. Наприклад, у статті [25] описаний командний студентський проект стосовно моделювання класичної задачі про аеропорт за допомогою складних структур даних.

Незважаючи на вивченість тематики складних структур даних, питання навчання студентів комп'ютерних спеціальностей складних структур даних недостатньо з'ясовані з педагогічної точки зору.

Призначення статті – описати підходи до навчання складних структур даних, формування у студентів комп'ютерних спеціальностей навичок сучасного мислення, а також дати рекомендації до структурування програми навчання з тематики, що стосується складних структур даних.

Бінарні дерева пошуку. Структура даних являє собою колекцію алгоритмів для зберігання та вилучення даних. Операції над структурами даних можна підрозділити на два типи – запити (queries), при опрацюванні яких не змінюється структура даних, і поновлення (updates), при виконанні яких змінюється структура даних, що може спричинити порушення раніше заданих властивостей. При побудові комп'ютерної системи важливим є продуманий вибір структури даних, за допомогою якої кожна задача за допомогою системи може бути розв'язана більш ефективно.

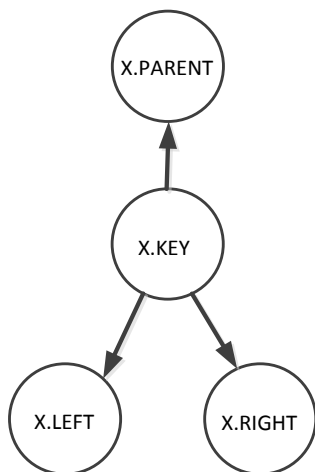


Рис. 2

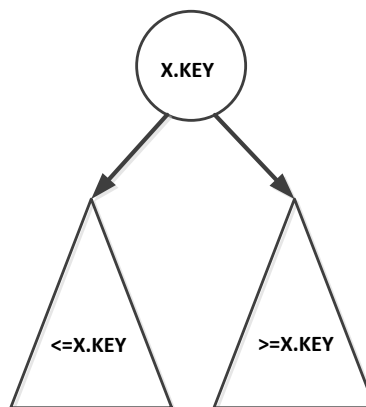


Рис. 3

Бінарні дерева пошуку (BST – binary search trees) являють собою досить складну структуру даних. Кожна вершина x BST містить не більше двох дочірніх вершин (ліва і права дочірні вершини), а також такі дані (див. рис. 1): $x.key$ – значення, збережене у вершині x ; $x.parent$ – вказівник на батьківську вершину (NIL у разі кореневої вершини, оскільки відсутня батьківська вершина); $x.left$ – вказівник на лівий дочірній вузол (NIL у разі відсутності лівого дочірнього вузла); $x.right$ – вказівник на правий дочірній вузол (NIL у разі відсутності правого дочірнього вузла). Крім того, для кожної вершини виконується властивість бінарного дерева: в лівому піддереві зберігаються ключі, які не

перевершують за значенням ключ, що міститься в даній вершині, в правому піддереві – навпаки, перевищують це значення (див. рис. 2).

Бінарні дерева пошуку (BST) використовуються під час виконання таких операцій: запити – BSTree-WALK (обхід дерева), MIN (знаходження мінімального за значенням елемента), MAX (знаходження максимального за значенням елемента), SUCCESSOR (знаходження наступного за значенням елемента), PREDECESSOR (знаходження попереднього за значенням елемента), SEARCH (пошук вузла із заданим значенням ключа), а також операції оновлення BST – INSERT (вставляння вузла), DELETE (вилучення вузла). При вивченні цих операцій необхідно звертати увагу студентів на особливості виконання цих операцій, а також особливості вставляння і вилучення вузлів. При вилученні вузлів необхідно розглянути всі три можливі сценарії вилучення. У таблиці 1 наведено поділ теми «Бінарні дерева пошуку» за результатами навчання, а також за рівнями освітніх цілей таксономії Блума [28].

Таблиця 1.

Тема	Результати навчання (що повинні вміти студенти)	Рівні навчальних цілей (таксономія Блума)
Бінарні дерева пошуку (BST)	Обов’язкова кількість годин – 4 Додаткові години для вивчення теми – 2	
1. Означення BST, властивість BST	– формулювати означення BST, властивості BST; – будувати BST.	Знання – К
2. Обхід дерева	– виводити всі ключі BST у відсортованому порядку; – реалізовувати алгоритм обходу дерева; – визначати час роботи за алгоритмом обходу дерева.	Знання – К, Застосування – А
3. Операції, які не змінюють структуру BST (запити): – мінімальне (MIN) та максимальне значення (MAX), – наступний (SUCCESSOR) і попередній (PREDECESSOR) елементи, – пошук елемента SEARCH (x).	– реалізовувати алгоритми операцій-запитів: MIN, MAX, SUCCESSOR, PREDECESSOR, SEARCH; – визначати час роботи за алгоритмами виконання операцій-запитів; – аналізувати гірші, кращі та середні варіанти можливої реалізації.	Застосування – А, Оцінювання – Е
4. Операції, в результаті виконання яких змінюється структура BST (оновлення): – вставляння вузла INSERT (x), – вилучення вузла DELETE (x).	– реалізовувати алгоритми операцій, за якими змінюють структуру BST: INSERT, DELETE; – реалізовувати три сценарії вилучення вузлів в BST; – визначати час роботи за алгоритмом виконання операцій, в результаті застосування яких змінюється структура BST; – аналізувати гірші, кращі та середні варіанти можливої реалізації.	Застосування – А, Аналіз – Ап, Оцінювання – Е

Збалансовані дерева пошуку. Під час вивчення бінарних дерев пошуку слід звернути увагу на важливість побудови збалансованого бінарного дерева пошуку. При цьому необхідно ознайомити студентів з видами бінарних дерев пошуку, за допомогою яких забезпечують збалансованість і гарантують прийнятний час роботи навіть у найгіршому випадку. У таблиці 2 наведено перелік збалансованих дерев пошуку. Також слід знайомити студентів із структурами даних, що не є бінарними деревами пошуку, але їх використання забезпечує швидке виконання основних операцій типу запитів і оновлень. Як приклад можна навести дерево ван Емде Боаса (van Emde Boas tree) – vEB-дерева.

Таблиця 2.

№	Вид дерева	Автори структури даних
1.	BST – Бінарне дерево пошуку (Binary Search Tree)	Незалежно відкриті багатьма дослідниками в кінці 1950-х років. Детально описані в книзі Д. Кнута [6].
Збалансовані дерева пошуку		
2.	RBT – Червоно-чорне дерево (Red-Black Tree)	Запропоновано в 1972 році Р. Байером в роботі [7]; в 1978 Л. Гібас та Р. Седжвік для балансування дерева в статті [8] запропонували концепцію кольорів.
3.	B-дерево (B-tree)	Запропоновано в 1972 році Р. Байером в роботі [7]
4.	AVL-дерево (AVL-tree)	Запропоновано Г.М. Адельсон-Бельським та Е.М. Ландісом в 1962 році, описані в роботі [8]. У кожному вузлі вводиться додаткове поле, в якому зберігається висота піддерева даного вузла.
5.	2-3-4 дерево (2-3-4 tree)	Запропоновано Р. Байером и Е. Мак Крейтом в 1972 році [9], [10].
6.	Дерево, що розширюється (Splay-tree)	Введені Д. Слейтором та Р. Гар'яном в 1985 році, описані в статті [11].
7.	Список з пропусками (Skip list)	Введені в 1989 році В. Пафом в статті [12].
8.	Дерево зі штрафами (Scaregoat tree)	Представляють собою самозбалансовані бінарні дерева пошуку. Вперше представлені А. Андерсоном в статті [13] у 1989 році. Повторно відкриті І. Гальперінім і Р. Рівестом в статті [14] у 1993 році.
9.	Декартове дерево (Treaps)	Введені Р. Сейделом і С. Арагоном, описані в статті [15] в 1989 році, уточнені в статті [16] в 1996 році.
Асоціативний масив		
10.	Дерево ван Емде Боаса (van Emde Boas tree) – vEB-дерева	Асоціативний масив, використання якого дозволяє зберігати цілі числа в діапазоні $[0; M)$, де M – число, що складається не більше ніж з k біт. Введені П. ван Емде Боасом в 1977 році. Описані автором у статтях [17], [18]. Не є бінарними деревами пошуку.

Слід приділити увагу зіставленню тимчасової складності основних операцій різних реалізацій складних структур даних. У таблиці 3 наведено характеристики даних за параметрами необхідної для зберігання пам'яті, часу виконання операцій пошуку (всі операції-запити виконуються за такий же час), вставлення і вилучення (дані взяті з процитованих в табл. 2 джерел). Можна запропонувати студентів до заповнення такої таблиці.

Таблиця 3.

Часова складність основних операцій						
Параметри порівняння	BST		RBT, AVL, B-дерево, Scaregoat tree, Treaps		vEB-дерева	
	Найгірший випадок	Середній випадок	Найгірший випадок	Середній випадок	Найгірший випадок	Середній випадок
Пам'ять	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(M)$	$O(M)$
Пошук	$O(n)$	$O(h)$	$O(\log n)$	$O(\log n)$	$O(\log \log M)$	$O(\log \log M)$
Вставлення	$O(n)$	$O(h)$	$O(\log n)$	$O(\log n)$	$O(\log \log M)$	$O(\log \log M)$
Вилучення	$O(n)$	$O(h)$	$O(\log n)$	$O(\log n)$	$O(\log \log M)$	$O(\log \log M)$
Позначення: n – кількість вершин, h – висота дерева, M – верхній діапазон інтервалу зберігання цілих чисел в vEB-дереві – число, що складається не більше ніж з k біт						

Червоно-чорні дерева. Тема «Червоно-чорні дерева (RBT)» є достатньо складною для вивчення студентами, але її розгляд важливий, оскільки дозволяє знайомити студентів зі складними структурами даних і прикладами збалансованих бінарних дерев. При цьому необхідно пояснювати студентам, де застосовуються RB-дерева. Також необхідно звернути увагу на те, що RB-дерево – це дерево бінарного пошуку, в якому є додаткове поле $x.\text{color}$ (червоний або чорний) і яке задовільняє червоно-чорні властивості (RB-властивості):

- кожна вершина – або червона, або чорна;
- корінь дерева – чорний;
- кожен лист ($nil[T]$) – чорний;
- якщо вершина – червона, то обидва дочірніх вузли – чорні; для кожного вузла всі шляхи від нього до листя, нащадків цього вузла, містять одну і ту саму кількість чорних вершин.

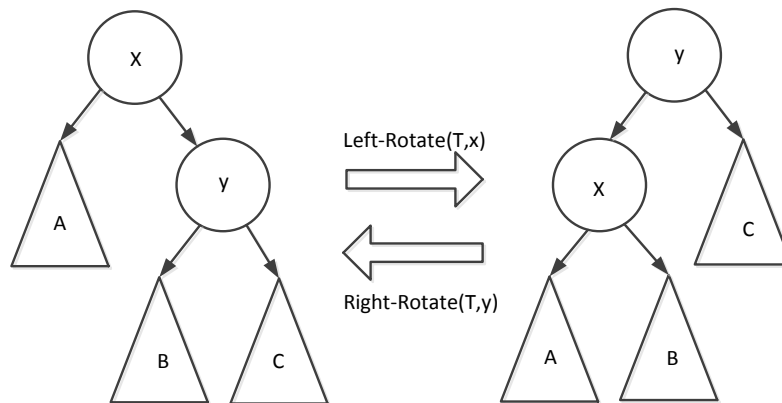


Рис. 4

За рахунок наявності поля `color` і RB-властивостей дерево може бути збалансованим. Операції-запити з використанням RB-дерев виконуються так само, як і з використанням BST. При застосуванні операцій, в результаті виконання яких змінюється структура дерева (вставляння INSERT і видалення DELETE), порушується RB-властивість і вимагається балансування. У разі використання RB-дерев для відновлення RB-властивості застосовується операція повороту – ROTATE. На рис. 3 показані результати виконання операції повороту лівого – Left-Rotate (T,x) – і правого – Right-Rotate (T,y) – поворотів в RB-дереві. У кодах операцій повороту вимагається перепозначення вказівників вузла.

У таблиці 4 наведено план вивчення теми «Червоно-чорні дерева», а також очікувані результати навчання із зазначенням рівнів освітніх цілей за таксономією Блума.

Таблиця 4.

Збалансовані бінарні дерева пошуку		
Тема	Результати навчання (що повинні вміти студенти)	Рівні навчальних цілей (таксономія Блума)
Червоно-чорні дерева (RBT)	Обов'язкова кількість годин – 4 Додаткові години для вивчення теми – 2	
1. Властивість RBT, чорна висота $bh(x)$	– надати означення RB-дерева; – описати червоно-чорні властивості; – визначити висоту RB-дерева; чорну висоту $bh(x)$; – оцінювати висоту RB-дерева, в якому є n внутрішніх вузлів.	Знання – К, Застосування – А, Аналіз – А _n
2. Повороти – LEFT-ROTATE (T, x), RIGHT-ROTATE (T, y)	– змінювати структуру вказівників за допомогою операції повороту; – реалізовувати алгоритми лівого і правого поворотів; – оцінювати час роботи за алгоритмом, реалізації поворотів.	Застосування – А
3. Вставляння вузла	– реалізовувати алгоритм вставляння вузла в RB-дерево; – проводити балансування після порушення однієї або кількох RB-властивостей; – знати інваріанти циклу алгоритму вставляння в RBT; – описувати варіанти дій у кожному з трьох можливих випадків, що виникають в ході реалізації алгоритму; – оцінювати час роботи за алгоритмом; – розрізнити операції вставляння в бінарному дереві пошуку і в червоно-чорному дереві.	Застосування – А, Аналіз – А _n , Оцінювання – Е

<p>4. Вилучення вузла</p>	<ul style="list-style-type: none"> – реалізувати алгоритм вилучення вузла в RB-дереві; – проводити балансування після порушення однієї або кількох RB-властивостей; – описувати варіанти дій у кожному з чотирьох можливих випадків, що виникають в ході реалізації алгоритму; – оцінювати час роботи за алгоритмом; – розрізняти операції вилучення в бінарному дереві пошуку і в червоно-чорному дереві. 	<p>Застосування – А, Аналіз – Ап, Оцінювання – Е</p>
---------------------------	---	--

AVL-деревя. Інший тип збалансованих дерев пошуку, який доцільно вивчати зі студентами, – AVL-деревя. Це бінарні деревя пошуку, які для балансування доповнені параметром висоти вузла (або зберігається значення різниці висот лівого і правого піддерев, тобто у вузлі може зберігатися одне із значень $-1, 0, +1$), а також підпорядковуються властивості AVL-деревя: значення лівого і правого піддерев будь-якого вузла не повинно відрізнятись більш, ніж на 1. На рис. 4 проілюстровано властивість AVL-деревя. На рис. 4 параметр різниці висот вузла ± 1 або 0, замінено стрілками.

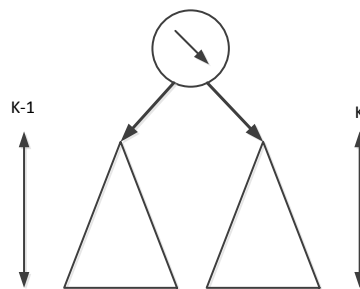


Рис. 5

Операції з AVL-деревями такі самі, як і з бінарними деревями пошуку BST, червоно-чорними деревями RBT. Пошук найменшого (MIN), найбільшого (MAX), наступного і попереднього за значенням елементів, пошук вузла із заданим ключем – виконуються за тими самими алгоритмами, що і в BST.

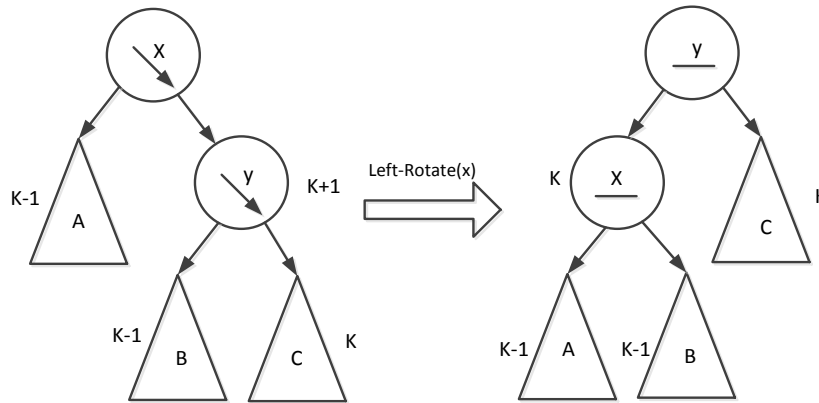


Рис. 6

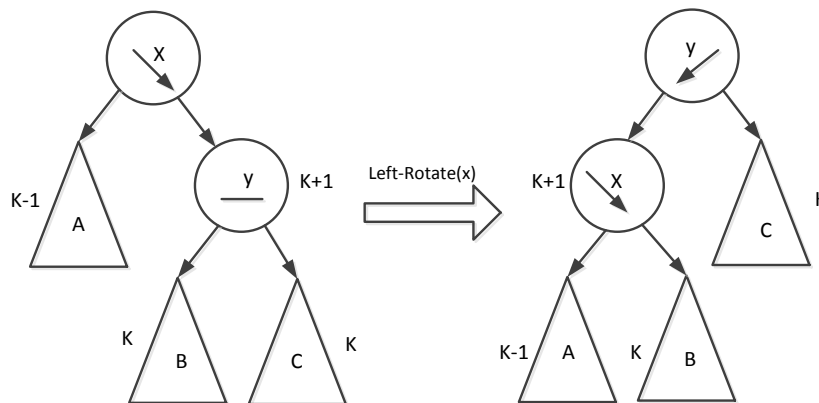


Рис. 7

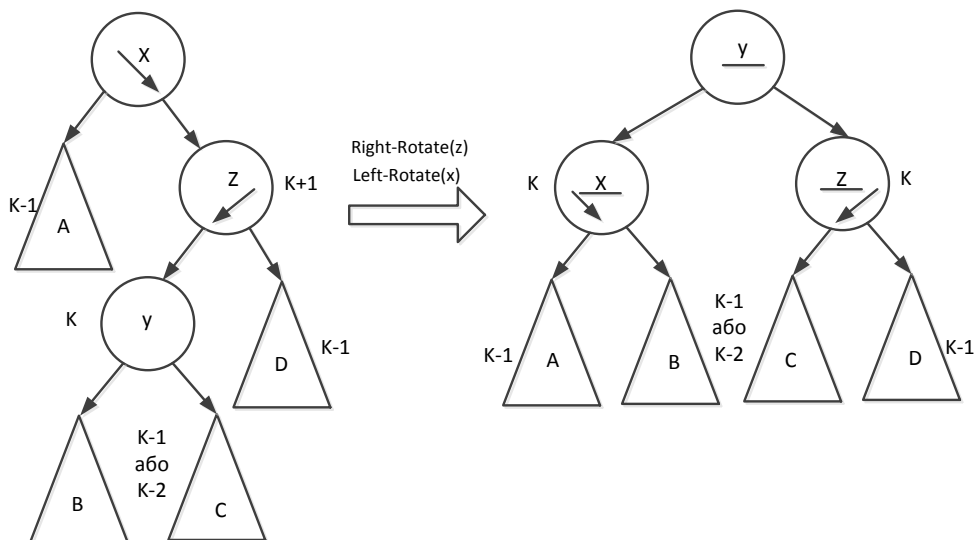


Рис. 8

Операції вставлення та вилучення вузла спочатку виконуються, як і в бінарному дереві пошуку, але потім необхідна перевірка, чи не порушено AVL-властивість. Дані необхідно провести балансування. Для балансування в AVL-деревях застосовуються повороти, можливі три різних випадки, які показані на рис. 5, рис. 6, рис. 7.

У таблиці 5 наведена тематика вивчення AVL-дерев, очікувані результати навчання, а також рівні освітніх цілей за Блумом. Оскільки тема AVL-дерев є завершальною, то рекомендується дати можливість студентам в рамках індивідуальної роботи провести порівняльний аналіз AVL та RBT структур.

Таблиця 5.

Збалансовані бінарні дерева пошуку		
Тема	Результати навчання (що повинні вміти студенти)	Рівні навчальних цілей (таксономія Блума)
AVL-деревья	Обов'язкова кількість годин – 4 Додаткові години для вивчення теми – 2	
1. Означення та загальні властивості AVL-деревя	– дати означення AVL-деревя; – формулювати властивість AVL-деревя; – визначити параметри балансу кожного вузла.	Знання – К
2. Балансування, обертання / повороти	– змінювати структуру вказівників за допомогою операції повороту; – реалізовувати алгоритми поворотів (три випадки); – оцінювати час роботи за алгоритмом, реалізації поворотів.	Застосування–А, Аналіз – Ап, Оцінювання – Е
3. Вставлення вузла та балансування	– реалізовувати алгоритм вставлення вузла; – проводити балансування після порушення AVL-властивості; – оцінювати час роботи за алгоритмом; – відрізнити операції вставлення в бінарному дереві пошуку і в AVL-дереві.	Застосування–А, Аналіз – Ап, Оцінювання – Е
4. Вилучення вузла та балансування	– реалізовувати алгоритм вилучення вузла; – проводити балансування після порушення AVL-властивості; – оцінювати час роботи за алгоритмом; – відрізнити операції вилучення в бінарному дереві пошуку і в AVL-дереві.	Застосування–А, Аналіз – Ап, Оцінювання – Е
5. Порівняння RBT та AVL дерев	– проводити порівняння означень RBT та AVL; – зіставляти операції балансування, вставлення вузла, вилучення; – порівнювати ефективність AVL і RB-дерев.	Аналіз – Ап, Оцінювання – Е

Висновки. Знання складних структур даних – важлива складова освіти інженера-програміста. На їх основі формується у студентів глибоке розуміння алгоритмів організації зберігання даних, проведення операцій з даними, а також вміння розробляти програмне забезпечення з урахуванням характеристик апаратного забезпечення і ефективності алгоритму.

Крім досить широкого процитованого в статті переліку джерел з тематики складних структур даних, при підготовці студентів комп'ютерних спеціальностей з дисципліни «Алгоритми і структури даних» основним підручником, безсумнівно, залишається класична книга авторів Томаса Кормена, Чарльза Лейзерсона, Рональда Ривеста, Кліффорда Штайна «Алгоритмы: построение и анализ» [29], [30].

Матеріал статті відображає багаторічний досвід навчання студентів Кримського інженерно-педагогічного університету дисципліни «Алгоритми і структури даних» за напрямом підготовки 6.040302 – Інформатика.

Список використаних джерел

1. Devaney L. The key skills today's employers desire / L. Devaney // eSchool News, vol. 16, #1, 2013. – P. 58-61.
2. Becker K. A tool for teaching advanced data structures to computer science students: an overview of the BDP system / K. Becker, M. Beacham // Journal of Computing Sciences in Colleges, v. 16, # 2, 2001. – P. 65-71.
3. Lawrence R. Teaching data structures using competitive games / R. Lawrence // IEEE Transactions on Education, v. 47, #4. – IEEE Press Piscataway, NJ, USA, 2004. – P. 459-466.
4. Сейдаметова З.С. Алгоритми та структури даних в підготовці інженерів-програмістів / З.С. Сейдаметова, С.Н. Сейтвелієва // Наук. часопис НПУ ім. М.П.Драгоманова. Сер. 2. Комп'ютерно-орієнтовані системи навчання. – К. : Вид-во НПУ ім. М.П. Драгоманова, 2011. – Вип. 11 (18). – С. 37-42.
5. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд. : Пер. с англ. : Уч. пос. / Д.Э. Кнут. – М.: Вильямс, 2000. – 832 с.
6. Bayer R. Symmetric binary B-Trees: Data structure and maintenance algorithms / R. Bayer // Acta Informatica, v. 1, #4, 1972. – P. 290-306.
7. Адельсон-Вельский Г. М. Один алгоритм организации информации / Г.М. Адельсон-Вельский, Е.М. Ландис. // Доклады АН СССР, 1962, n. 146, № 2. – С. 263–266.
8. Guibas L.J. A dichromatic framework for balanced trees / L. J. Guibas, R. Sedgwick // Proceedings of the 19th Annual Symposium on Foundations of Computer Science. – IEEE CS Washington, DC, USA, 1978. – P. 8-21.
9. Bayer R. Organization and maintenance of large ordered indexes / R. Bayer, E. McCreight // Acta Informatica, v. 1, #3, 1972. – P. 173-189.
10. Sleator D. D. Self-Adjusting Binary Search Trees / D.D. Sleator; R.E. Tarjan // Journal of the ACM, v. 32, # 3, 1985. – P. 652–686.
11. Pugh W. Skip lists: a probabilistic alternative to balanced trees / W. Pugh // Communications of the ACM, v. 33, # 6, 1990. – P. 668–676.
12. Andersson A. Improving partial rebuilding by using simple balance criteria / A. Andersson. // Proc. Workshop on Algorithms and Data Structures. – Springer-Verlag, 1989. – P. 393-402.
13. Galperin I. Scapegoat trees / I. Galperin, R.L. Rivest. // Proceedings of the 4th annual ACM-SIAM Symposium on Discrete algorithms. – Society for Industrial and Applied Math. Philadelphia, PA, USA, 1993. – P. 165-174.
14. Aragon C. Randomized Search Trees / C.R. Aragon, R. Seidel // Proc. 30th Symp. Foundations of Computer Science. – Washington, D.C.: IEEE Computer Society Press, 1989. – P. 540-545.
15. Seidel R. Randomized Search Trees / R. Seidel, C.R. Aragon // Algorithmica, vol. 16, #4/5, 1996. – P. 464-497.
16. Emde Boas P. Preserving order in a forest in less than logarithmic time and linear space / P. van Emde Boas // Information Processing Letters, # 6(3) 1977. – P. 80-82.
17. 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.
18. Štrbac-Savić S. Comparative performance evaluation of the AVL and red-black trees / S. Štrbac-Savić, M. Tomašević // Proceedings of the Fifth Balkan Conference in Informatics. – ACM New York, NY, USA, 2012. – P. 14-19.
19. Okasaki C. Alternatives to Two Classic Data Structures / C. Okasaki // Proceedings of the 36th SIGCSE technical symposium on Computer science education. – ACM New York, NY, USA, 2005. – P. 162-165.
20. Visser W. Test Input Generation for Red-Black Trees using Abstraction / W. Visser,

C.S. Pasareanu, R. Pelanek // Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. – ACM New York, NY, USA, 2005. – P. 414-417.

21. Pfaff B. Performance Analysis of BSTs in System Software / B. Pfaff // ACM SIGMETRICS Performance Evaluation Review, v. 32, #1, 2004. – ACM New York, NY, USA, 2004. – P. 410-411.

22. Sen S. Deletion without rebalancing in balanced binary trees / S. Sen, R.E. Tarjan // Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms. – Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2010. – P. 1490-1499.

23. Darga P.T. Efficient software model checking of data structure properties / P.T. Darga, C. Boyapati // Newsletter ACM SIGPLAN Notices - Proceedings of the 2006 OOPSLA Conference, v. 41, #10, 2006. – ACM New York, NY, USA, 2006. – P. 363-382.

24. Balasa F. Using Red-Black Interval Trees in Device-Level Analog Placement with Symmetry Constraints / F. Balasa, S. C. Maruvada, K. Krishnamoorthy // Proceedings of the 2003 Asia and South Pacific Design Automation Conference. – ACM New York, NY, USA, 2003. – P. 777-782.

25. Dake P. Designing a collaborative cross-campus airport (or other transit) simulation project / P. Dake, S. Khoja, R. Bryant, G. Orr // Journal of Computing Sciences in Colleges, v. 24, #2, 2008. – P. 72-73.

26. Clements A.T. Scalable address spaces using RCU balanced trees / A.T. Clements, M.F. Kaashoek, N. Zeldovich // Newsletter ACM SIGARCH Computer Architecture News – ASPLOS '12, v. 40, # 1, 2012. – ACM New York, NY, USA. – P. 199-210.

27. Ellen F. Non-blocking binary search trees / F. Ellen, P. Fatourou, E. Ruppert, F. van Breugel // Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing. – ACM New York, NY, USA, 2010. – P. 131-140.

28. Anderson L.W. A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives / L.W. Anderson, D.R. Krathwohl, B.S. Bloom. – New York: Addison Wesley Longman, 2001. – 352 p.

29. 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.

30. 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.

Струтинська О.В.

НПУ імені М. П. Драгоманова

Особливості формування компетентностей у галузі дистанційного навчання інформатичних дисциплін майбутніх учителів інформатики

Сучасне суспільство орієнтоване на широке використання інформаційно-комунікаційних технологій. У зв'язку з цим застосування інформаційно-комунікаційних технологій у шкільній освіті постійно розширюється. Тому нагальною постає проблема підготовки майбутніх вчителів до широкого використання інформаційно-комунікаційних технологій у своїй професійній діяльності.

Крім того, останнім часом у навчальний процес багатьох вищих навчальних закладів і шкіл активно впроваджуються дистанційні технології навчання. Це зумовлює необхідність вирішення питання підготовки фахівців у галузі дистанційного навчання.

На основі аналізу досліджень [3] та власного досвіду можна зробити висновки, що в Україні жоден вищий навчальний заклад не готує майбутніх вчителів до навчання відповідних предметів за дистанційними технологіями. Основними проблемами у цій галузі залишаються такі:

- недосконалість існуючої законодавчої бази у галузі дистанційного навчання;
- не розроблені програми підготовки вчителів (викладачів) дистанційного навчання різних предметів;
- не визначений правовий статус вчителя (викладача) дистанційного навчання, не розроблені його кваліфікаційні характеристики, відсутні методики підготовки, норми навчального навантаження тощо;
- не визначена уніфікована структура побудови дистанційних курсів;
- не розроблені механізми охорони авторських прав і захисту дистанційних курсів від розповсюдження дидактичних та методичних матеріалів в мережі Інтернет без згоди авторів.
- недостатня увага для вирішення цього питання приводить до недостатнього фінансування, і, як наслідок, поганого технічного оснащення навчальних закладів.

На даний час в Україні існують лише курси підвищення кваліфікації у галузі дистанційного навчання, в основному орієнтовані на вчителів середніх загальноосвітніх навчальних закладів.

Останнім правовим документом, прийнятим у галузі дистанційного навчання, є Наказ Міністерства освіти і науки України від 25.04.2013 № 466 "Положення про дистанційне навчання". У