

Прокоп Ю.В.

Одеська національна академія зв'язку імені О.С. Попова

Трофименко О.Г.

Національний університет «Одеська юридична академія»

Дикий О.В.

Національний університет «Одеська юридична академія»

ДОСЛІДЖЕННЯ ПІДХОДІВ ДО ВИКЛАДАННЯ КУРСУ «АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ» ДЛЯ СТУДЕНТІВ ІТ-СПЕЦІАЛЬНОСТЕЙ

Досліджено світові практики викладання в університетах світу курсу CS2 як аналогу курсу «Алгоритми та структури даних» (АСД), що є базовим під час підготовки студентів галузі 12 «Інформаційні технології» в Україні. Аналіз наповнення курсу в різних зарубіжних та вітчизняних університетах показав доволі широкий спектр змістовних тем, але їх набори, поєднання та послідовність розміщення суттєво відрізняються. З'ясовано перелік тем за рейтингом того, наскільки часто вони зустрічаються у курсі в різних вишах світу. Він такий: рекурсія, алгоритми пошуку і сортування, оцінювання швидкості алгоритмів, списки, черги, стеки, дерева, купа, графи, карти і хеш-таблиці тощо. Виявлено, що в українських університетах послідовність тем у аналогічному переліку суттєво відрізняється від загальносвітових підходів. Нерідко в навчальних програмах відсутні теми рекурсії, купи і хеш-таблиць. Крім того, є явно виражений часовий перекид у бік вивчення алгоритмів, а вивченню структур даних приділяється менше уваги. Аналіз труднощів засвоєння матеріалу курсу студентами показав, що здебільшого погано сприймаються теми рекурсії і зв'язаних списків. З'ясовано, що вибір мови програмування для цього курсу залежить від регіональних тенденцій. У США університети використовують Java, Python, C++ і C. В європейських університетах – C++, C і Java. Понад 2/3 українських університетів у курсі АСД використовують мову програмування C++; серед інших мов – Python, Delphi, Java або C#. Проаналізований спектр проблем і специфіки викладання курсу АСД для студентів ІТ-спеціальностей показав актуальність пошуку нових підходів для кращого засвоєння його здобувачами вищої освіти і потребу подальшого дослідження з урахуванням регіональної специфіки.

Ключові слова: алгоритми та структури даних, CS2, ІТ-освіта, мова програмування.

Постановка проблеми. Курс «Алгоритми та структури даних» (АСД) є одним із базових у закладах вищої освіти в Україні під час підготовки студентів галузі 12 «Інформаційні технології». В університетах світу цей курс здебільшого має назву CS2. У курсі вивчаються основні етапи проектування, розроблення та аналізу алгоритмів, досліджується ефективність алгоритмів, динамічні структури даних і алгоритми на графах тощо. Набуті навички сприятимуть написанню ефективного робочого коду у розробленні нових проєктів. Нині на співбесідах ІТ-компаній та під час оцінки кандидатів найму часто питання з алгоритмів рішення задач (Problem Solving) є ключовими. «Алгоритмічні» питання можуть становити від 60% до 100% завдань у співбесіді. Здебільшого такі завдання полягають у неочевидному вирішенні, а не у реалізації, і дають

змогу з'ясувати здатність фахівця до розроблення нового функціоналу програмного продукту та оптимізації наявного, що дасть змогу і далі просувати компанії вгору.

Важливість курсу АСД у процесі підготовки ІТ-фахівців змушує викладачів в університетах нашої країни та всього світу приділяти значну увагу вдосконаленню його викладання та шукати нові підходи для кращого засвоєння його здобувачами вищої освіти. Адже розуміння принципів роботи алгоритмів і структур даних під час розроблення програмного забезпечення дає змогу покращити продуктивність програм, поліпшити якість коду і прискорити його роботу, а тому є фундаментальним для становлення майбутніх програмістів. Алгоритми використовуються практично в усіх сферах інформаційних технологій: у криптографії, під час аналізу текстів, зображень

і відео, в біоінформатиці, аналізі великих даних, штучному інтелекті, задачах оптимізації тощо.

Важливість результативного вивчення АСД зумовлює актуальність гнучкого коригування змістовного наповнення дисципліни, перегляду її навчальної програми для приведення у відповідність до сучасних вимог ІТ-галузі, а також пошуку підходів викладання щодо покращення ефективності засвоєння матеріалу здобувачами вищої освіти.

Аналіз останніх досліджень и публікацій. Дискусії про способи викладання базових дисциплін із програмування на першому і другому курсах навчання бакалаврату під час підготовки ІТ-фахівців (у термінології зарубіжних університетів – CS1 і CS2) не вщухають вже декілька десятиліть. Зокрема, увагу привертають роботи, які обговорюють структуру та наповнення дисциплін [1–5], мови програмування, використовувани в вивченні курсів [6–10], і складнощі, з якими стикаються студенти під час сприйняття матеріалу [4, 11, 12]. Нині більшість досліджень у цій сфері присвячені аналізу викладання курсу CS2 в університетах Європи та США, а тому цей досвід варто переосмислити крізь призму регіональної специфіки ІТ-освіти в Україні з метою залучення успішних підходів та можливості внесення змін у робочі програми АСД в українських вишах.

Постановка завдання. Метою статті є дослідження світових практик викладання курсу CS2 (АСД) в університетах з метою визначення кола найактуальніших розділів дисципліни, вивчення різних підходів до викладання, виявлення питань і завдань, що викликають найбільші труднощі у студентів під час вивчення, а також порівняння цих складників із тим, як викладається курс АСД в українських університетах.

Виклад основного матеріалу дослідження.

1. Аналіз підходів до викладання CS2 в університетах світу

У зарубіжних університетах не існує єдиного підходу до змісту курсу CS2. Здебільшого цей курс присвячений саме алгоритмам і структурам даних, проте в частині університетів у межах CS2 вивчають засоби об'єктно-орієнтованого програмування (ООП). Цим зумовлені певні складнощі з аналізом підходів до викладання зазначеного курсу. У статті йтиметься тільки про перший варіант, коли CS2 є прототипом курсу АСД.

Курс CS2, як і більшість курсів бакалаврату ІТ-освіти, має чітко визначений набір передумов, необхідних для його успішного вивчення. Зазвичай для нього потрібне знання курсів з основ

програмування (CS1) та дискретної математики. Деякі розділи CS2 потребують знання певних розділів вищої математики.

У міжнародному дослідженні [4], яке проводиться з метою розроблення концептуального опису курсу CS2, беруть участь викладачі з усього світу. Проміжні результати дослідження дозволили сформулювати перелік тем, що вважаються найважливішими і найчастіше вивчаються у курсі CS2. До них належать: рекурсія, алгоритми пошуку і сортування, оцінювання швидкості алгоритмів, списки, черги, стеки, дерева, купа, графи, карти і хеш-таблиці. Однак вибір змістовного наповнення цього курсу, як і більшості інших, значною мірою залежить від самого університету і певного викладача, а тому набір розділів перетинається в різних університетах лише частково, не існує жодної теми, що вивчається в цьому курсі в усіх без винятку університетах.

Зазвичай під час вивчення структур даних використовується один із двох підходів: або з використанням бібліотек і класів, або із самостійною реалізацією структур даних та подальшим їх використанням [4].

Дискусійним питанням є вибір мови програмування для курсу CS2. Згідно з дослідженням [6], 42% університетів США в цьому курсі використовують мову програмування Java, 28% – Python, 16% – C++, 5% – C. Причому понад 50% вишів, які використовували Java і C++ в курсі CS1, використовують ту саму мову і в CS2. За даними [7], найпопулярнішою для вивчення структур даних є мова Java, її частка становить 34,8%. Використовуються й інші мови, наприклад, C++ [13] та C# [14]. Дослідження [8] з'ясувало, що в європейських університетах, включно з Україною, в другому семестрі першого курсу навчання використовують мови програмування: C++ (32,8%), C (28,9%) і Java (16,1%). На другому курсі в третьому і четвертому семестрі використовують відповідно мову: Java в 39% і 35,1%, C – 19,5% і 13,5%, C++ – 17% і 19,6%, а C# – 6% і 14,2%.

Вибір тієї чи іншої мови програмування для CS2 зумовлений насамперед тематичним наповненням курсу – АСД або ООП, а також підходом до використання бібліотек або до навчання власної реалізації структур даних. В останньому разі для курсу CS2 важливими є вказівники та інструменти керування пам'яттю, яких немає в явному вигляді в мовах Java і Python, а тому доречною є мова C++, яка володіє цими засобами [9]. Крім того, під час вибору мови програмування як для CS1, так і для CS2 важливим є питання безболіс-

ного переходу від одного курсу до іншого. Дослідження [9] виявило складність для розуміння студентами структури даних «зв'язані списки», що могло бути наслідком вивчення Python у курсі CS1 через відсутність у цій мові вказівників та засобів маніпулювання пам'яттю. Подібна складність виникає і після вивчення в курсі CS1 мови Java, що підтверджено дослідженням [10].

Курс CS2 є доволі складним для розуміння студентами, багато хто з них сприймає матеріал лише частково [15]. За даними [9], невдачі наприкінці цього курсу становлять 5,7%. Серед найскладніших тем курсу дослідники називають рекурсію, купу, алгоритми сортування та пошуку, аналіз швидкості алгоритмів, зв'язані списки, черги, карти і хеш-таблиці, рекурсивне оброблення дерев [4, 11, 12]. Причому перелік найскладніших тем CS2 суттєво відрізняється у різних дослідників, що може свідчити про наявність суб'єктивних чинників, таких як: розподіл часу за темами, добір завдань для лабораторних робіт, попередній досвід вивчення програмування у студентів, педагогічна майстерність викладача тощо.

2. Аналіз підходів до вивчення курсу АСД в Україні

На основі даних сайту vstup.info було складено список закладів вищої освіти України, в яких готують студентів за спеціальностями галузі знань 12 «Інформаційні технології». Далі на сайтах цих навчальних закладів було вибрано та проаналізовано силабуси, навчальні програми та/або матеріали для студентів, що є у відкритому доступі. Насамперед з'ясувалися і порівнювалися такі аспекти:

- курс АСД є самостійним курсом чи він є частиною іншого курсу;
- в якому семестрі він викладається;
- кількість кредитів;
- використовується мова програмування;
- використання бібліотек або самостійна реалізація структур даних;
- розділи, що наповнюють курс АСД.

Загалом до розгляду було взято 50 передових вишів України. З'ясовано, що традиційно курс АСД є незалежним, однак у деяких освітніх програмах він є складовою частиною курсу «Алгоритмізація та програмування» (АП), здебільшого це стосується спеціальності 122 «Комп'ютерні науки». Найпоширенішим є викладання цього курсу в третьому, рідше – у другому семестрі бакалаврату. Середня кількість кредитів – 4.

Найчастіше курс АСД складається з двох умовних частин, присвячених вивченню відповідно

алгоритмів (пошуку, сортування тощо) і структур даних (зв'язаних списків, стеків, черг, дерев, графів). Як і в зарубіжних університетах, перелік тем вивчення значно відрізняється в різних вишах України. Особливо помітними є відмінності у часових пропорціях вивчення алгоритмів та структур даних у цьому курсі. В частині університетів виявлено відчутний перебік у бік алгоритмів (зважаючи на кількість годин та тематику). У деякі програми теми з вивчення структур даних включено лише номінально або взагалі відсутні. Недостатня увага до викладання структур даних суттєво відрізняє програми курсу АСД в українських вишах від закордонних.

Аналіз 22 навчальних програм виявив 35 різних тем курсу АСД. Загальними темами, які входять до не менш ніж 10 навчальних програм дисципліни, є: базові алгоритми сортування (бульбашкове, вибором, вставками), базові структури даних (зв'язані списки, черги, стеки), алгоритми пошуку в числових послідовностях (зокрема, бінарний пошук), бінарні дерева, вдосконалені алгоритми сортування (Шелла, Хоара тощо), зовнішнє сортування, аналіз алгоритмів, загальні поняття теорії алгоритмів, алгоритми на графах, рекурсія. Ці теми названо за зменшенням частоти згадувань і без урахування кількості годин, що відводяться на їх вивчення. Інші теми мають набагато меншу кількість перетинів у навчальних програмах дисципліни АСД різних вишів. Порівнюючи ці показники із закордонними, можна дійти висновку, що деякі поширені в іноземних університетах теми (наприклад, рекурсія і алгоритми текстового пошуку) значно рідше зустрічаються в програмах курсу АСД у вишах України. Деякі теми (зокрема, купи і хеш-таблиці) відсутні в понад половині навчальних програм курсу.

Традиційно вважається, що перша частина курсу АСД (алгоритми) практично не залежить від мови програмування. Деякі університети дозволяють студентам виконувати її лабораторні роботи засобами будь-якої відомої їм мови, а в лекціях нерідко для пояснень використовують псевдокод. Друга частина курсу АСД (структури даних) вже суттєво залежить від того, яку мову взято за основу. Понад 2/3 українських університетів у курсі АСД використовують мову програмування C++. Серед інших мов у деяких навчальних програмах застосовують Python, Delphi, Java або C#. Мова програмування Delphi, як і C++, має в явному вигляді вказівники. Водночас решта названих мов їх не мають, що може ускладнювати розуміння реалізації динамічних структур даних.

Так само, як і в зарубіжних університетах, чимало українських вишів обмежуються використанням бібліотек і не вимагають від студентів вивчення самостійної реалізації структур даних. Такий підхід переважно притаманний програмам, де виявлено перекид у бік вивчення алгоритмів.

Як показує досвід викладання в Одеському державному університеті інтелектуальних технологій і зв'язку, найбільші складності у студентів викликають зв'язані списки і бінарні дерева. З'ясовано, що причиною цього є поверхневе знайомство зі вказівниками у курсі АП, який передусім дисципліні АСД. Водночас теми, пов'язані з алгоритмами пошуку і сортування, сприймаються студентами краще за інші. Це зумовлено значною увагою до вивчення засобів опрацювання масивів у курсі АП, завдяки чому теми, пов'язані з ними, надалі здаються студентам більш легкими і зрозумілими. Можливим підходом до вирішення проблем з опанування складних тем є збільшення годин на вивчення відповідних розділів, викладання за принципом від пояснення простих задач до більш складних, перегляд кількості та змісту завдань у лабораторних роботах, а також розбиття індивідуальних варіантів для виконання студентами за рівнями складності.

Висновки. Дослідження різних підходів до викладання курсу CS2 в зарубіжних університетах та аналогічної дисципліни АСД в українських вишах дало змогу з'ясувати, що цей курс для студентів є складним, із високим рівнем невдач на іспитах і відсівом. Зібрано та проаналізовано перелік основних тем наповнення цієї дисципліни в різних закладах вищої освіти. Аналіз навчальних програм дисципліни АСД в українських вишах виявив переважне вико-

ристання мови C++ і помітний перекид у бік вивчення саме алгоритмів під час викладання курсу. Встановлено, що серед використовуваних мов програмування у зарубіжних вишах преважує Java, далі за популярністю використання у цьому курсі йдуть мови Python та C/C++. Незалежно від методики і підходів викладання у різних університетах світу більшу частину розділів курсу студенти називають складними для розуміння, найчастіше труднощі викликають теми рекурсії, купи, зв'язаних списків тощо. Зважаючи на з'ясовані проблеми, завдання пошуку способів для підвищення успішності під час вивчення курсу залишаються вельми актуальними і потребують подальшого дослідження, при цьому варто враховувати регіональну специфіку. Адже цей курс є фундаментальним у підготовці якісних ІТ-фахівців. Знання та навички, що здобуваються студентами під час його вивчення, закладають фундаментальну базу для подальшого навчання програмування в університеті та майбутньої професійної діяльності. Успішність вивчення цього курсу суттєво впливає на засвоєння інших предметів з вивчення засобів розроблення програмного забезпечення і значною мірою пов'язана з подальшим становленням майбутнього фахівця ІТ-галузі. Здобувачі ІТ-освіти мають багато вчитися і під час навчання, і під час професійної діяльності, щоби встигати за змінами в ІТ-індустрії загалом та новими трендами у вузькоспеціалізованих напрямках. Хоча популярність ІТ-освіти зростає, майбутнім студентам варто зважати на потребу певних зусиль і самостійного навчання щодо вдосконалення теоретичних і практичних навичок із розроблення програмного забезпечення.

Список літератури:

1. Farghally M.F., Koh K.H., Ernst J.V., Shaffer C.A. Towards a concept inventory for algorithm analysis topics. *In Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 2017. P. 207–212. DOI: 10.1145/3017680.3017756.
2. Porter L., Zingaro D., Nam Liao S., Taylor C., Webb K., Lee C., Clancy M. BDSI: A Validated Concept Inventory for Basic Data Structures. *In Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER '19)*. Association for Computing Machinery, New York, NY, USA, 2019. P. 111–119. DOI: 10.1145/3291279.3339404.
3. Porter L., Zingaro D., Lee C., Taylor C., Webb K., Clancy M. Developing Course-Level Learning Goals for Basic Data Structures in CS2. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. Baltimore, MD, USA, 2018. P. 858–863. URL: <https://www.cs.swarthmore.edu/~kwebb/papers/LearningGoals-SIGCSE.pdf>.
4. Wittie L., Kurdia A., Peng J., Kelly J., Huggard M. Developing a Concept Inventory for Computer Science 2: What should it focus on and what makes it challenging? *IEEE Frontiers in Education Conference (FIE '20)*. 2020. P. 1–8. URL: http://www.tara.tcd.ie/bitstream/handle/2262/92988/CS2_Concept_Inventory_FIE_2020.pdf;jsessionid=BD4D1BE6611256FB0C6A3CF76556F59C?sequence=1.

5. Ібрагімова Л.А. Аналіз змісту робочих програм з дисципліни «Алгоритми та структура даних» у закладах вищої освіти. *Збірник наукових праць «Педагогічні науки»*. 2019. Вип. LXXXVI. С. 252-256. DOI: 10.32999/ksu2413-1865/2019-8648.
6. Siegfried R., Herbert-Berger K., Siegfried J. CS2 and the Impact of Programming Language Choice. *51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. New York, NY, USA, 2020. P. 1353. DOI: 10.1145/3328778.3372644.
7. Silva D., Aguiar R., Dvconlo D., Silla C. Recent Studies About Teaching Algorithms (CS1) and Data Structures (CS2) for Computer Science Students. *IEEE Frontiers in Education Conference (FIE'19)*. Covington, KY, USA, 2019. P. 1-8. DOI: 10.1109/FIE43999.2019.9028702.
8. Aleksić V., Ivanović M. Introductory Programming Subject in European Higher Education. *Informatics in Education*. 2016. Vol. 15, No. 2. P. 163-182. DOI: 10.15388/infedu.2016.09.
9. Layman L., Song Y., Guinn C. Toward Predicting Success and Failure in CS2: A Mixed-Method Analysis. *ACM Southeast Conference (ACM SE '20)*. New York, NY, USA, 2020. P. 218-225. DOI: 10.1145/3374135.3385277.
10. Davies S., Polack-Wahl J., Anewalt K. A snapshot of current practices in teaching the introductory programming sequence. *42nd ACM technical symposium on Computer science education (SIGCSE '11)*. New York, NY, USA, 2011. P. 625-630. DOI: 10.1145/1953163.1953339.
11. Zingaro D., Taylor C., Porter L., Clancy M., Lee C., Nam L., Webb K. Identifying student difficulties with basic data structures. *In Proceedings of the 14th ACM Conference on International Computing Education Research*. 2018. P. 169-177. URL: <https://www.cs.oberlin.edu/~ctaylor/pubs/icer143-zingaroA.pdf>.
12. Tenenberg J., Murphy L. Knowing what I know: An investigation of undergraduate knowledge and self-knowledge of data structures. *Computer Science Education*. 2005. Vol. 15, No. 4. P. 297-315. DOI: 10.1080/08993400500307677.
13. Herala A., Vanhala E., Nikula U. Object-oriented programming course revisited. *In Proceedings of the 15th Koli Calling Conference on Computing Education Research (Koli Calling '15)*. Association for Computing Machinery, New York, NY, USA, 2015. P. 23-32. DOI: 10.1145/2828959.2828974.
14. Krpan D., Rosić M., Mladenovic S. Teaching Basic Programming Skills to Undergraduate Students. *Proceedings of CIET'14*. 2014. P. 147-158. URL: https://www.researchgate.net/publication/275271925_Teaching_Basic_Programming_Skills_to_Undergraduate_Students.
15. Yeomans L., Zschaler S., Coate K. Transformative and Troublesome? Students' and Professional Programmers' Perspectives on Difficult Concepts in Programming. *ACM Transactions on Computing Education*. Vol. 1, No. 1. P. 1-27. DOI: 10.1145/3283071.

Prokop Yu.V., Trofymenko O.H., Dykyi O.V. RESEARCH OF APPROACHES TO TEACHING THE COURSE “ALGORITHMS AND DATA STRUCTURES” FOR COMPUTER SCIENCE STUDENTS

The article studies the world practices of teaching the CS2 course in the universities as an analogue of the "Algorithms and Data Structures" course (ADS), which is the basic one for training students in Ukraine. The analysis of the course content in various foreign and domestic universities showed a wide range of content topics, but their sets, combinations, and sequence of placement differ significantly. The list of topics is ranked according to how often they occur in the course in different universities around the world. It is as follows: recursion, search and sorting algorithms, algorithm speed estimation, linked lists, queues, stacks, trees, heap, graphs, maps, and hash tables, etc. It was revealed that in Ukrainian universities the sequence of topics in a similar list differs significantly from global approaches. Often in the curriculum, there are no topics of recursion, heap, and hash tables. Also, some Ukrainian universities have revealed a significant time bias towards studying algorithms, and less attention is paid to data structures. Analysis of the difficulties of mastering the course material by students showed that mostly the topics of recursion and linked lists are poorly perceived. It was found that the choice of programming language for this course depends on regional trends. In the US, universities use Java, Python, C++, and C. European universities use C++, C, and Java. More than 2/3 of Ukrainian universities in the ADS course use the C++ programming language; other languages include Python, Delphi, Java, or C#. The analyzed range of problems and specifics of teaching the course of ADS for students of IT specialties showed the relevance of finding new approaches for better mastering by higher education students and the need for further research taking into account regional specifics.

Key words: algorithms and data structure, CS2, IT education, programming language.