НАВЧАЛЬНА ПРОГРАМА
ТРЕНІНГУ ВЧИТЕЛІВ ІНФОРМАТИКИ «ПРОГРАМУВАННЯ НА МОВІ С++, ПІДГОТОВКА ДО ОЛІМПІАД З ПРОГРАМУВАННЯ»

(30 годин)

Укладачі:

С. Д. Вапнічний, викладач Ужгородського національного університету

В.А.Ребрина, ст.викладач кафедри теорії та методик природничо-математичних дисциплін і технологій

О.В.Міца, завідувач кафедри інформаційних управляючих систем та технологій Ужгородського національного університету

Рецензенти:

П.М.Григорук, професор кафедри автоматизованих систем і моделювання в економіці Хмельницького національного університету, доктор економічних наук

С.С.Петровський, доцент кафедри комп'ютерних наук та інформаційних технологій проектування Хмельницького національного університету, кандидат педагогічних наук

ПОЯСНЮВАЛЬНА ЗАПИСКА

Олімпіадні змагання заохочують дітей до навчання і сприяють глибокому вивченню відповідного предмета. Крім цього, участь в олімпіадах з інформатики розвиває такі якості, як спритність і нестандартність мислення. Навіть найлегше завдання або найпростіший алгоритм двоє програмістів реалізують абсолютно по-різному. Реалізація придуманих ідей − це своєрідний спосіб їх викладення за допомогою однієї з дозволених на олімпіаді мов програмування, яка є тільки інструментом. Використовуючи однакові інструменти: літери і слова людської мови; практично неможливо знайти хоча б двох письменників, які зможуть створити однаковий твір на задану тему. Таку аналогію можна провести і з іншими творчими професіями: художники, скульптори, музиканти ...

Багато колишніх олімпійців організовують успішні проекти, пов'язані не тільки з програмуванням та IT-технологіями. Завдяки участі в олімпіадах, їм вдалося виробити стійкість до складних психологічних навантажень. Провівши тисячі годин часу за тренуваннями, намагаючись не витратите навіть один з них даремно, вони навчилися оцінювати вірогідність перемоги і поразки. Освоїли

існуючі і виробили власні методи боротьби зі стресовими ситуаціями, з сумнівами і занепокоєннями, які відчувають олімпійці під час змагань в тій чи іншій мірі. Це вироблена здатність приймати важливі рішення і нести відповідальність за них.

Як і в спорті або в будь-який інший сфері діяльності, щоб досягти успіху в олімпіадному програмуванні, потрібно дуже багато працювати. Уміння придумувати розв'язки, генерувати ідеї - такі ж, як грати в футбол, бігати марафони, малювати картини ... І що найголовніше - всі ці навички можна тренувати і розвивати. Так, талант або схильність до певного виду діяльності відіграють не останню роль у становленні людини, як професіонала. Ці якості однозначно полегшують формування і розвиток спеціальних здібностей, необхідних в обраній сфері діяльності.

Але відсутність цих вроджених якостей ніяк не може перешкодити розвиватися і перемагати на олімпіадах з програмування. Значно більший вплив на результат мають такі людські якості, як дисциплінованість, витримка і стійкість. Напевно, все ж більшість тих, хто читає, погодяться, що останні якості можна успішно розвивати. Що і роблять школярі і студенти, готуючись до наступних олімпіад. По суті, перемога на змаганні не так важлива і цікава сама по собі. Для учасника це тільки показник свого особистісного розвитку - чи зміг він побороти свою лінь, організувати свій вільний час, в кінці кінців, чи став він краще щодо попередньої версії себе.

Напевно, головні навички, які прививає участь в олімпіадах - це власне програмування і вміння шукати помилки в написаних програмах. Існують різні формати проведення та оцінювання робіт на олімпіадах. Але в кожної з них кількість отриманих балів за запропонований розв'язок безпосередньо залежить від правильності останнього. Тому потрібно реалізувати ідею без помилок, бажано з першого разу. В іншому випадку учасник повинен швидко виявити і виправити неточності. Уміння налагоджувати програми - одне з найважливіших умінь в світі розробці.

Розроблений модуль «ПРОГРАМУВАННЯ НА МОВІ С++, ПІДГОТОВКА ДО ОЛІМПІАД З ПРОГРАМУВАННЯ» включає в себе основні розділи і поняття

з олімпіадного програмування, а також дозволяє краще засвоїти мову програмування високого рівня С++. Знання та практичний досвід, що будуть отримані в процесі вивчення курсу, дозволять значно розширити кругозір та можливості слухачів.

СТРУКТУРА МОДУЛЮ ПІДВИЩЕННЯ КВАЛІФІКАЦІЇ ВЧИТЕЛІВ ІНФОРМАТИКИ «ПРОГРАМУВАННЯ НА МОВІ С++, ПІДГОТОВКА ДО ОЛІМПІАД З ПРОГРАМУВАННЯ»

№ З/п	Теми занять	Кількість годин		
		Всього	лекції	практичні
1.	Рекурсія. Перебір з поверненням (`BackTracking`)	3	1	2
2.	Алгоритм Евкліда. Решето Ератосфена.	3	1	2
3.	Бінарний та тернарний пошуки	3	1	2
4.	Структури даних. Бібліотека STL.	3	1	2
5.	Дерево відрізків. Дерево Фенвіка	3	1	2
6.	Динамічне програмування.	3	1	2
7.	Жадібні алгоритми.	3	1	2
8.	Графи. Подання графів. Алгоритми обходів графа	3	1	2
9.	Графи. Шляхи у графі.	3	1	2
10.	Залікова робота	3	1	2
	ВСЬОГО :	30	10	20

ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ЗА ТЕМАМИ:

Тема 1. Рекурсія. Перебір з поверненням (`BackTracking`).

План лекційного заняття.

1. Поняття рекурсивної функції.
2. Рекурсивні процедури та функції.
3. Приклади рекурсивних алгоритмів
4. Перебір з поверненням (`BackTracking`).
5. Приклади задач на `BackTracking` .

Тема 2. Алгоритм Евкліда. Решето Ератосфена.

Лекція, 2 години

План лекційного заняття.

1. Знаходження дільників числа. Факторизація числа.
2. Алгоритм Евкліда для знаходження НСД двох чисел. НСК двох чисел.
3. НСД та НСК для n-чисел
4. Функція Ейлера.
5. Решето Ератосфена.

Тема 3. Бінарний та тернарний пошук.

План лекційного заняття.

1.Ідея бінарного пошуку.
2. STL `lower_bound` та `upper_bound`.
3. Бінарний пошук по відповіді.
4. Ідея тернарного пошуку.
5. Знаходження мінімальної відстані від точки до відрізка.
6. Метод золотого перерізу.

Тема 4. Структури даних. Бібліотека STL.

План лекційного заняття.

1. Опис структури. Обробка типу структура Робота із структурами.

2. Динамічні масиви. Основні методи заголовкового файлу <vector>.
3. Стек. Черга. Множина.
4. Особливості подання множин у C++: заголовкові файли <set>, <multiset>.
5. Основні функції директив: erase(), find(), lower_bound().

Тема 5. Дерево відрізків. Дерево Фенвіка.

План лекційного заняття.

1. Формулювання задачі `Range Minimum/Sum Query`
2. Ідея дерева відрізків, аналіз обчислювальної складності
3. Приклади реалізації (зверху вниз та знизу вверх)
4. Операції модифікації на проміжку
5. Загальна ідея дерева Фенвіка. .
6. Пояснення бітових операцій. Приклад реалізації
7. Мінімум на відрізку Модифікації на проміжку
8. Спуск по дереву Фенвіка

Тема 6. Динамічне програмування.

План лекційного заняття.

1. Загальна ідея динамічного програмування
2. Поняття стану та переходів
3. Динамічне програмування "вперед" і "назад"
4. Відновлення відповіді в задачах ДП
5. Приклади задач.

Тема 7. Жадібні алгоритми. Практичне заняття, 2 години

План лекційного заняття.

1. Поняття жадібного алгоритму.
2. Приклади задач.
3. Умови застосування жадібного алгоритму

Тема 8. Графи. Подання графа. Алгоритми обходів графа.

Практичне заняття, 2 години

План практичного заняття (2 год.)

1. Основні поняття теорії графів
2. Способи представлення графу
3. Алгоритм BFS
4. Алгоритм DFS
5. Знаходження мостів та точок сполучення в графі

Тема 9. Графи. Шляхи у графах.

Практичне заняття, 2 години

План практичного заняття (2 год.)

1. Поняття шляху в графі.
2. Алгоритм Дейкстри.
3. Алгоритм Флойда.

Тема 10.Заліікова робота.

Практичне заняття, 3 години

План практичного заняття (3 год.)

Залік (у формі розв'язання задач на контесті).

ПІДСУМКОВА АТЕСТАЦІЯ СЛУХАЧІВ

Підсумкова атестація слухачів проходить у вигляді практичної контрольної роботи.

Орієнтовна тематика індивідуальних творчих робіт

1. Ізоморфне представлення задач в інформатиці
2. Діофантові рівняння. Розширений алгоритм Евкліда.
3. Формула Піка.
4. Задача про розташування ферзів на шаховій дошці.

5. Задача про розташування радарів.
6. Задача обходу конем всіх кліток шахової доски.
7. Метод динамічного програмування. Задача про ранець.
8. LCS-алгоритм знаходження найбільшої загальної підпослідовності.
9. Алгоритми знаходження найбільшої зростаючої підпослідовності.
10. Динамічне програмування за профілем.
11. Поняття про ‘жадібні’ алгоритми. Задачі про здачу, про розклад, планування виробничої лінії, вибір заявок та інші.
12. Загальні відомості про графи. Основні означення та типи задач.
13. Метод Форда-Беллмана знаходження найкоротшої відстані між вершинами графа.
14. Метод Флойда знаходження найкоротшої відстані між вершинами графа.
15. Метод Дейкстри знаходження найкоротшої відстані між вершинами графа.
16. Метод Левіта, його застосування та реалізація.
17. Пошук в глибину, його застосування та реалізація.
18. Пошук в ширину, його застосування та реалізація.
19. Декартове дерево.
20. Методи сортування масивів.
21. Дерево Фенвіка. Приклади та застосування.
22. Дерево відрізків. Приклади та застосування.
23. Система неперетинаючихся множин.
24. Метод “решета Ератосфена” знаходження простих чисел.
25. Поняття NP-повноти. Основні класи задач, що відносяться до NP-повних.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Берж К. Теория графов и ее применение. - М.: ИЛ, 1962. – 201 с.
2. Беров В.И., Лапунов А.В., Матюхин В.А., Пономарев А.Е. Особенности национальных задач по информатике. Киров: Триада-С, 2000. – 183 с.
3. Вирт Н. Алгоритм+структуры данных=Программы. - М.: Наука, 1989. – 281 с.
4. Грузман М. Эвристика в информатике. – Винница: ВПК "Артур", 1998. – 579 с.
5. Кирякин С., Ляпунов М., Окулов Н. Международные олимпиады с информатики 1989–1996 гг. – М.: BHV, 1996. – 536 с.
6. Кнут Д. Искусство программирование. Том 1. – М.: Мир, 1999. – 892 с.
7. Кнут Д. Искусство программирование. Том 2. – М.: Мир, 1999. – 827 с.
8. Кнут Д. Искусство программирование. Том 3. – М.: Мир, 1999. – 903 с.
9. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 2001. – 957 с.
10. Кристофидес Н. Теория графов. Алгоритмический подход.– М.: Мир, 1978. – 498 с.
11. Препарата Ф., Шеймос М.Вычислительная геометрия: введение. - М.: Мир, 1989. – 382 с.
12. Скиена С., Ревилла М. Олимпиадные задачи по программированию. Руководство по подготовке к соревнованиям. − М.: Кудиц-Образ, 2005. – 672 с.
13. Шень А. Решение олимпиадных задач с информатики. – М.: Мир, 1997. – 632 с.
14. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Построение и анализ вычислительных алгоритмов. − М.: Мир, 1979. – 592 с.
15. Грэхем Р., Кнут Д., Паташник О. Конкретная математика, ISBN 978-5-8459-1588-7, «ВИЛЬЯМС», 2013. − 784 с.
16. Горошко Ю. , Інформаційне моделювання у підготовці майбутніх учителів математики та інформатики. Чернігів, Україна: Видавець Лозовий В. М., 2012. – 424 с.

17. Мельник В. Задачник з програмування. Київ, Україна: Видавничий дім ”Шкільний світ”, 2006. – 212 с.
18. Літня школа з програмування: Матеріали лекцій, умови та розбір задач / За ред. Олександра Міци, Сергія Оришича. – Ужгород: Видавництво “ФОП Сабов А.М.”, 2017. – 276 с.
19. Оршанский С. О решении олимпиадных задач по программированию формата ACMICPC. – М.: Мир ПК - диск, №9, 2005.
20. Жуковський С. Аналіз, дослідження та розв’язування конкурсних задач під час учнівської олімпіади з інформатики // Інформаційні технології в освіті, №5, 2010. − С. 152-159.
21. Ребрина В.А. Олімпіади з інформатики: завдання, ідеї та коди розв’язків. 8-11 класи.– Харків: Вид-во «Ранок», 2018.– 160 с.