

Предложено подходы к соответствующей подготовке будущих учителей информатики – систематичность, поэтапность, непрерывность. В связи с этим предложено формы применения технологий облачных вычислений: как средства организации учебно-познавательной деятельности, как объекта изучения, как средства разработки информационно-образовательных ресурсов. Особое внимание уделено проектным методикам организации обучения в процессе применения академических облаков.

Ключевые слова: учитель информатики, компетентность, облачные технологии.

THE TRAINING MODEL OF FUTURE COMPUTER SCIENCE TEACHERS TO APPLICATION OF CLOUD TECHNOLOGIES

Yu. Ramskyi, V. Oleksiuk

Resume. The article explores the concept of using cloud computing during the training process of future computer science teachers. The author defines the scientific and methodical principles of formation of future computer science teachers' professional competencies in the field of cloud computing – continuity and graduality. The research deals with the main components of the social and professional competency of computer science teachers. The author paid special attention to the project methods of teaching cloud technologies.

Keywords: computer science teacher, competence, cloud technologies.

DOI 10.31392/NPU-nc.series2.2018.20(27).05

УДК 373.5.016:004

Ю.В. Горошко¹, В.І. Мельник², О.В. Міца³

¹доктор педагогічних наук, професор

Національний університет «Чернігівський колегіум» імені Т.Г. Шевченка

²Заслужений учитель України,

Кременчуцький педагогічний коледж ім. А.С. Макаренка

³кандидат технічних наук, доцент,

ДВНЗ «Ужгородський національний університет»

ПРО ТУРНІРИ ЮНИХ ІНФОРМАТИКІВ

Анотація. У статті проаналізовано Всеукраїнські турніри юних інформатиків за останні роки. Розглянуто особливості проведення кожного з трьох турів турніру. Показано позитивну динаміку зростання кількості команд з 2013 року. Наведено розбір задач третього туру турніру 2017 р. З'ясовано, що серед проблем щодо турніру важливою є відсутність офіційного онлайн-ресурсу турніру, тому пропонується організувати відповідний сайт для збереження даних про турніри – умови задач як першого, так і третього турів, учнівські розв'язки задач першого туру, підсумкові таблиці з результатами тощо.

Ключові слова: турнір юних інформатиків, комп'ютерна програма, моделювання, АСМ.

Потреба у спеціалістах з ІТ у сучасному світі невідносно зростає. Зрозуміло, що підготовка таких фахівців повинна починатися ще в школі, як на уроках інформатики так і в позакласній роботі. Одним із напрямів зацікавлення учнів сучасними ІТ-технологіями, профорієнтаційної роботи у цьому напрямку, пошуку та стимулювання обдарованих учнів, формування в них інформатичних компетентностей є турнір юних інформатиків.

Турнір юних інформатиків започатковано з ініціативи Тетяни Караванової, доцента Чернівецького національного університету імені Ю. Федьковича, зараз заступника голови журі турніру. Уперше проведено у 2002 р. у місті Чернівці.

Турнір є командним, причому команди можуть бути різновіковими. Робота з різновіковими групами є особливо ефективною для таких творчих заходів, як олімпіади та турніри [1].

Очна частина турніру проходить у три тури. Для першого туру готується проблемна задача. У більшості випадків це прикладна задача, розв'язування якої вимагає від учнів поглиблених знань із різних предметів: фізики, математики, хімії тощо. Розв'язком цієї задачі повинна бути комп'ютерна програма (програмний комплекс). Як виняток, у турнірі 2016 р. для першого туру було запропоновано написати комп'ютерну гру-бій, причому програми різних команд повинні були брати участь у таких віртуальних боях (автор задачі Андрій Паньків). Ці віртуальні бої викликали велику цікавість та азарт у більшості учасників. Саме в цьому турі дається найбільше балів і перевіряється вміння команди аналізувати складні тематичні завдання та подавати розв'язки у формі, близькій до впровадження.

Захист розробленої програми проходить у першому очному турі, причому кожна команда виступає у трьох ролях – представника розробленого проекту, опонента та рецензента. Слід відзначити певні труднощі для команд саме в поданні проекту, оскільки досить часто в такому поданні не показуються сильні сторони проекту, розробники зупиняються на другорядних. На цей

факт повинні звернути увагу перш за все тренери команд. Це саме стосується і документації до розроблених програмних проектів.

На другому очному етапі командам пропонується написати комп'ютерну програму за певний час. Як правило, це комп'ютерна гра за специфікаціями та наведеним прикладом від журі. Потім відбувається захист цієї програми командою. На виконання завдання другого туру дається обмежена кількість часу, в основному це 5 годин. У своєму розпорядженні команда має два комп'ютери, які можуть використовуватися для обміну даними. Тому для якісної роботи потрібний грамотний поділ завдання на частини і злагоджена робота членів команди. Команди, які проводять тренування в такому режимі, виступають суттєво краще.

Третій етап полягає у розв'язуванні командами алгоритмічних задач з опонуванням відповідних алгоритмів вибраною мовою програмування за правилами АСМ. Важливу роль в успішному виступі на третьому турі відіграють системні заняття з розв'язання задач алгоритмічного плану [2].

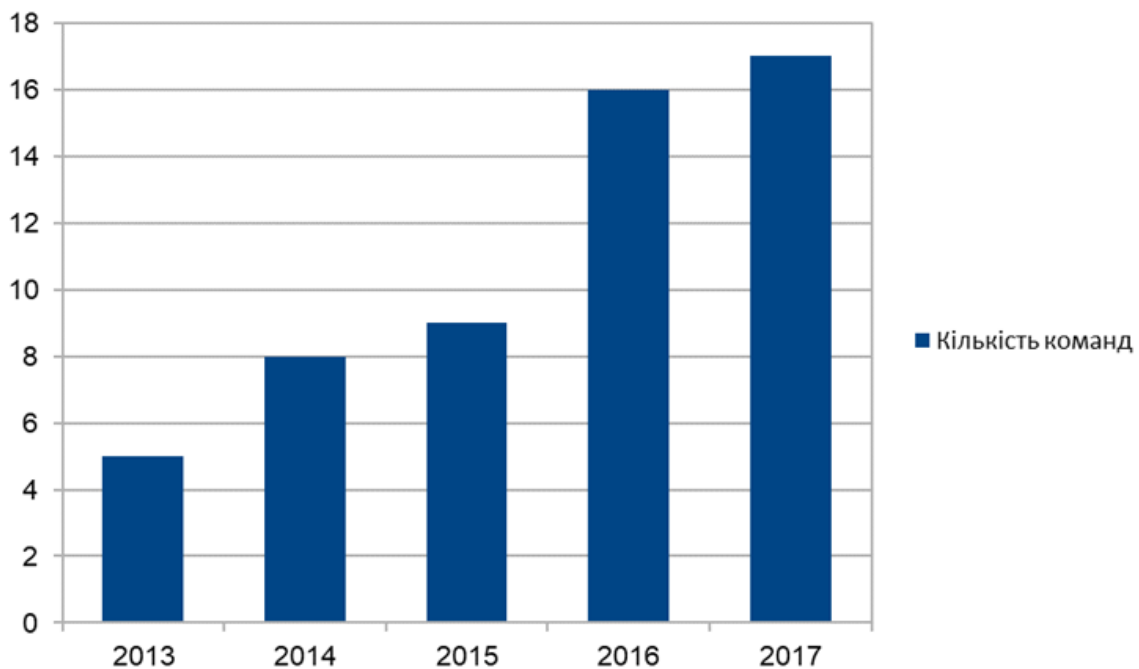
В останні роки спостерігається стрімке зростання інтересу до турніру юних інформатиків (див. Таблиця 1 та Діаграма 1).

Таблиця 1

<i>Рік</i>	<i>Кількість команд</i>
2013	5
2014	8
2015	9
2016	16
2017	17

Особливо слід відмітити значне зростання кількості команд з Харківської області та їх стабільно високі результати. Також слід відзначити команду Закарпатської області, яка за останні чотири роки отримала два третіх та два перших місця.

Слід відмітити високий рівень програмної підготовки учнів, які беруть участь у турнірі. Використовуються сучасні потужні середовища розробки програм такі, як, наприклад, Visual Studio, графічні бібліотеки, наприклад Unity3D, використовуються різні операційні системи, у тому числі і мобільні, організовується зв'язок з web-сервісами.



Діаграма 1

У 2017 р. XVI Всеукраїнський турнір юних інформатиків проходив з 15 по 19 листопада в м. Чернівці. Головою журі у цьому році був Юрій Васильович Горошко, професор Національного університету "Чернігівський колегіум" імені Т.Г. Шевченка, а експертом-консультантом Олександр Васильович Міца, доцент Ужгородського національного університету. У турнірі взяли

участь 17 команд. Як звичайно, турнір проходив у три тури. Перший тур – це домашнє завдання, яке виконувалося впродовж понад трьох місяців. Воно полягало в розробці моделі розрахунків електричних ланцюгів, розробці візуального конструктора електричних схем, розробці лабораторного фізичного практикуму для навчальних закладів та розпізнаванні електричних схем (автор задачі – Станіслав Петрович Мінаков). У другому турі потрібно було дослідити зразок комп'ютерної гри "Водопровід" та створити свою версію гри. У третьому турі потрібно було розв'язати 7 алгоритмічних задач за правилами АСМ (автор задач – Валентин Іванович Мельник заслужений учитель України, учитель інформатики Кременчуцького педагогічного коледжу ім. А.С. Макаренка).

У першому турі перемогла команда "Racoons" (Закарпатська обл.), у другому турі – "AC/DC" (Харківська обл.), а в третьому турі – знову перше місце здобула команда "Racoons" (Закарпатська обл.).

Більш детальні відомості про результати змагання наведено нижче:

Команда	I тур	II тур	III тур	Сума	
Racoons (Закарпатська обл.)	170,92	60,50	90,00	321,42	I
"AC/DC" (Харківська обл.)	147,75	81,00	75,00	303,75	II
"XLV.NEXT" (Харківська обл.)	150,92	71,33	75,00	297,25	II
"CuriosITy" (Харківська обл.)	154,17	59,17	75,00	288,33	II
"Паніка" (УФМЛ)	139,50	51,67	75,00	266,17	III
Збірна команда Волинської обл.	143,09	47,42	75,00	265,51	III
"CMD-3" (Харківська обл.)	141,25	46,25	78,00	265,50	III
"Політ" (Полтавська обл.)	133,25	53,17	75,00	261,42	III
"Gang of Botans" (Харківська обл.)	125,42	53,25	75,00	253,67	
"Core 47" (Харківська обл.)	130,58	57,08	65,00	252,67	
"Рекурсивні капібари" (м. Київ)	136,25	40,33	76,00	252,58	
"ScriptKiddies" (Чернівецька обл.)	152,58	31,00	65,00	248,58	
"3GMax" (Полтавська обл.)	118,17	38,42	65,00	221,58	
"Team zero" (Хмельницька обл.)	137,83	8,33	75,00	221,17	
"Кодери" (Житомирська обл.)	120,42	21,17	75,00	216,58	
Збірна команда «Педагогічний лицей і Вікторія-II» Кіровоградської області	114,83	36,17	35,00	186,00	
"RIMBA" (Хмельницька обл.)	106,17	24,00	55,00	185,17	

З деталями змагання та фотозвітотом можна ознайомитись на [3].

Наведемо аналіз задач третього туру змагань.

Задача А. Мультиграма

Леді – пристрасна любителька загадок. Найновіший тип загадок, з якими вона зіткнулася, вимагає від неї перевірити, чи є це слово мультиграмою.

Мультиграма – це слово, яке складається з об'єднання двох або більше слів, які є взаємними анаграмами. Перше з цих слів називається коренем мультиграми.

Наприклад, слово **bbabab** – це мультиграма з коренем **bba**, оскільки вона складається з анаграм **bba** та **bab**.

Допоможіть Леді розгадати загадку, визначивши, чи є задане слово мультиграмою, і визначте його корінь, якщо воно є мультиграмою. Якщо є кілька можливих коренів мультиграми, виведіть найкоротший. Будь ласка, зверніть увагу: два слова є взаємними анаграмами, якщо одне з них може бути отримане з іншого внаслідок зміни порядку символів.

Формат вхідних даних

Перший і єдиний рядок містить слово довжиною не більше 100 000 маленьких англійських літер.

Формат вихідних даних

Виведіть -1, якщо це слово не є мультиграмою. В іншому випадку виведіть найкоротший корінь цього слова в один рядок.

Пояснення

Пояснення першого прикладу: зверніть увагу, що слово "aa" також може бути коренем, але "a" коротше.

Пояснення другого прикладу: слово не є мультиграмою, оскільки "a" та "b" не взаємні анаграми.

Приклади вхідних та вихідних даних

Введення	Виведення
aaaa	a
ab	-1
bbabab	bba

Аналіз. Зауважимо, що довжина кореня повинна бути дільником довжини самої мультиграми. Таких дільників не буде дуже багато. Тому можна перебрати всі можливі довжини, і потім перевірити, чи підходить такий корінь.

Як перевірити? Нехай довжина кореня – K . Порахуємо кількість входжень кожної літери в перших K літерах слова. Тоді будемо брати кожні наступні K літер і дивитись, чи цей підрядок складений з таких самих літер, як і корінь (перші K літер слова).

Асимптотика рішення. Така перевірка буде здійснюватися за $N+D*(N/K)$, де D – розмір алфавіту, тобто 26. Усього таких перевірок буде не багато, адже різних дільників у числа не більше, ніж $2*\sqrt{N}$, де \sqrt{N} – корінь числа N . Тож середня асимптотика – $O(N*\sqrt{N})$.

Задача В. Шоу зайців-стрибунців

Троє зайців-стрибунців грають на числовій прямій. Кожен із них займає місце в одній із цілочисельних точок. Один із двох зайців, які розташовані зовні, стрибає в довільну вільну цілочисельну точку, між двома іншими. Ні в якому разі два зайці не можуть займати одну і ту саму точку.

Написати програму, за якою визначається кількість стрибків, яку можуть зробити зайці-стрибунці у найдовшій можливій грі.

Формат вхідних даних

Дано три цілих числа A, B, C ($0 < A < B < C < 100$) - початкові позиції зайців на числовій прямій.

Формат вихідних даних

Вивести у першому рядку кількість стрибків, яку можуть зробити зайці-стрибунці у найдовшій можливій грі.

Приклади вхідних та вихідних даних

Введення	Виведення
2 3 5	1
3 5 9	3

Аналіз. Зайці завжди стрибають раціонально, тож подивимось, як можуть змінитися позиції зайців після першого кроку. 1 варіант: заєць на позиції C стрибає на позицію $B-1$, якщо це можливо. Після цього заєць на позиції B стрибає в позицію $B-2$, якщо це можливо. Потім заєць із позиції $B-1$ стрибає на позицію $B-3$, і так далі. У такому випадку відповідь $B-A-1$. Це оптимальний варіант, якщо почне стрибати заєць із номером C . 2 варіант: заєць із позиції A переміщується на позицію $B+1$, потім заєць із позиції B переміщується на позицію $B+2$, і так далі. У цьому випадку відповідь $C-B-1$. Потрібно лиш узяти кращий варіант із цих двох, тобто максимум з $B-A-1$ та $C-B-1$.

Задача С. Решето Ератосфена

Ця задача є досить відомою, тому її умову та аналіз наводити не будемо.

Задача D. Дороги

Дорожня мережа в країні складається з N міст і M односторонніх доріг. Міста пронумеровані від 1 до N . Для кожної дороги відомі міста, які через неї з'єднуються, а також довжина дороги.

Вважатимемо, що дорога F є продовженням дороги E , якщо місто, до якого веде дорога E , є таким, з якого починається дорога F .

Шлях від міста A до міста B складається з набору таких доріг, що початком першої дороги є місто A , кожна дорога є продовженням тієї, що знаходиться перед нею, а місто останньої дороги, до якого вона веде, – місто B .

Довжина шляху – це сума довжин усіх доріг у ньому.

Шлях від A до B – це найкоротший шлях, якщо не існує іншого шляху від A до B коротшої довжини.

Завдання полягає в тому, щоб для кожної дороги виводити, скільки існує різних найкоротших шляхів через цю дорогу, за модулем 1 000 000 007.

Формат вхідних даних

В першому рядку міститься два цілих числа N, M ($1 \leq N \leq 1500, 1 \leq M \leq 5000$) - кількість міст і кількість доріг.

В кожному із наступних M рядків містяться три натуральних числа O, D, L – одностороння дорога від міста O до міста D довжини L . Номери O та D будуть різними, а L буде не більше 10000.

Формат вихідних даних

Вивести M цілих чисел, кожне в окремому рядку: для кожної дороги – кількість усіх найкоротших шляхів через неї її, за модулем 1 000 000 007.

Порядок цих номерів повинен відповідати порядку доріг у вхідних даних.

Приклади вхідних та вихідних даних

Введення	Виведення
4 3	3
1 2 5	4
2 3 5	3
3 4 5	

Аналіз. Нехай вершина K є початком шляху. Скористаємося алгоритмом Дейкстри, щоб знайти мінімальну відстань від цієї вершини до кожної іншої. Будемо зберігати ці відстані в масиві $dist$. Припустимо, кінцем шляху є вершина C . Визначимо, які ребра з неї входять до найкоротшого шляху. Нехай e ребро з C в E , тоді $dist[C]=dist[E]+len(C,E)$, де len – довжина шляху між вершинами. Перебираємо кожне з цих ребер, і також збільшуємо $kl[E]$. $kl[E]$ – кількість шляхів, що проходять через цю вершину. Зауважимо, що кожному елементу масиву kl потрібно спочатку надати значення 1, адже маршрут може закінчитися в будь-якій вершині. Далі потрібно застосувати динамічне програмування для підрахунку для кожного ребра кількості разів, коли воно використовується.

Асимптотика розв'язку. Якщо написати швидкий алгоритм Дейкстри, за $M \log N$, то розв'язок буде досягатися за $O(N^2 \log N)$, бо цей алгоритм застосовуватиметься N разів.

Задача Е. Тортики

Рубік і Мостік дуже люблять вечорами пити чай з тортиком. Тому вони кожен день заходять у магазин неподалік і купують там смачний торт із вишеньками й печенючками. Друзі дуже відповідально ставляться до вибору торта і вже запланували, які саме тортики вони хочуть у найближчі N днів. Рубік хоче, щоб в день i на тортіковій було не менше A_i і не більше B_i вишеньок. Мостік хоче, щоб у день i на тортіковій було не менше C_i і не більше D_i печенючок. Інших вимог до торта у Рубіка і Мостіка немає, зокрема Рубіку байдуже, яка кількість печенючок, а Мостіка не хвилює кількість вишеньок на тортіковій. Оскільки хлопці дуже дружні, то в день i вони куплять торт, якщо він сподобається хоча б одному з них.

Ви – власник магазину тортів і планує в найближчі N днів випускати один і той самий торт, тобто не міняти кількість вишеньок і кількість печенючок. Завдання – знайти такі значення для кількості вишеньок і кількості печенючок на торті, щоб Рубік і Мостік купили його у якомога більшу кількість днів.

Формат вхідних даних

У першому рядку міститься єдине число N ($1 \leq N \leq 200\,000$) – число днів, на яке Рубік і Мостік вже побудували плани.

У наступних N рядках знаходяться четвірки цілих невід'ємних чисел A_i, B_i, C_i, D_i – побажання друзів. Усі ці числа не перевищують 10^9 .

Формат вихідних даних

Виведіть два невід'ємних цілих числа P, Q – кількість вишеньок і печенюшок на тортіковій в будь-якому з оптимальних варіантів. Виведені числа не повинні перевищувати 10^9 .

Приклад вхідних та вихідних даних

Введення	Виведення
5	5 1
2 3 1 2	
3 5 1 2	
7 8 8 9	
5 6 2 3	
3 5 6 8	

Аналіз. Визначимо кількість вишеньок на тортіку. Цю підзадачу вирішимо за допомогою методу Сканлайн. Відсортуємо окремо всі початки й кінці відрізків $A[i]$ та $B[i]$, вважаючи, що кожен обмеження на вишеньки – це відрізок чисел, які можна взяти до відповіді. Тоді коли в черзі стоїть $A[i]$, додаємо цей відрізок, а якщо $B[i]$ – вилучаємо. Таким чином, можна знати для кожного $A[i]$ та $B[i]$, скільки відрізків покривають цю точку, що аналогічно до кількості тортів, які одразу

додаються до відповіді. Тепер визначимось щодо печеньок. Для цього будемо використовувати дерево відрізків. Потрібно вміти додавати якесь число в дерево відрізків, а також знаходити максимум у всьому дереві і номер вершини з максимумом. Тепер, якщо зафіксовано певну кількість вишеньок, візьмемо вершину в дереві відрізків із максимальним значенням, що і буде відповіддю щодо кількості печеньок. Але в цьому випадку деякі тортики зарахуються двічі. Тому коли отримуємо за допомогою методу Скайлайн новий відрізок, у дереві відрізків віднімемо 1 для всіх K , $C[i] \leq K \leq D[i]$, аби не зарахувати його двічі. А коли відрізок за допомогою методу Скайлайн вилучається, знову додамо 1. Таким чином зможнапорахувати максимальну кількість, не враховуючи жоден відрізок двічі.

Асимптотика розв'язку. Для відшукування розв'язку використовується сортування й дерево відрізків, тому розв'язок отримується за $O(N \log N)$.

Задача F. Акція

У книжковому магазині пропонується акційна пропозиція «Візьміть 3 книги, платіть за 2 найдорожчих». Таким чином, кожен клієнт, який обирає 3 книги, найдешевшу отримує безкоштовно. Звичайно, клієнти можуть брати більше книг і, залежно від того, як книги скласти по три в групу, одержують найдешевшу в кожній групі безкоштовно.

Наприклад, нехай ціна книжок, вибраних замовником, становить: 10 3 2 4 6 4 9. Якщо він організовує їх у групи: (10, 3, 2), (4, 6, 4) і (9), він отримає книгу ціною 2 за першу групу безкоштовно, і книгу ціною 4 з другої групи. Очевидно, що він не отримає нічого безкоштовно від третьої групи, оскільки в ній лише одна книга.

Леді, яка працює в книжковому магазині, має добрі наміри, і вона завжди хоче максимально знизити загальну вартість книг для кожного клієнта. За даними книжкових цін допоможіть леді організувати книги в групи так, щоб загальна вартість, яку повинен заплатити клієнт, була мінімальною.

Зверніть увагу: Леді не потрібно організовувати книги в групи, так що кожна група містить рівно 3 книги, але кількість книг у групі повинна бути від 1 до 3 включно.

Формат вхідних даних

В першому рядку міститься ціле число N ($1 \leq N \leq 100\,000$) – кількість книг, вибраних клієнтом. В кожному з наступних N рядків міститься одне ціле число C_i ($1 \leq C_i \leq 100\,000$) – ціна кожної книги.

Формат вихідних даних

Вивести мінімальну загальну вартість.

Пояснення

Пояснення першого прикладу: Леді може поставити книги ціною 3, 2, 2 в одній групі, і тільки книгу, яка коштує 3, в іншій групі, що є найдешевшою комбінацією.

Пояснення другого прикладу: Леді кладе книги вартістю 6, 4, 5 в одній групі, а 5, 5, 5 в іншій, що дає найдешевшу комбінацію.

Приклад вхідних та вихідних даних

Введення	Виведення
4	8
3	
2	
3	
2	

Аналіз. Візьмемо суму цін усіх книжок. Від неї треба відняти найбільш можливу суму цін книжок, які підуть за умовою безкоштовно. Найкраще кожного разу брати якнайбільшу ціну книжки за акцією, аби відповідь стала меншою. Тож якщо відсортувати всі ціни книжок за спаданням, кожен третій візьмемо безкоштовно, взявши її в групу з двома попередніми. Таким чином, отримаємо оптимальний варіант.

Асимптотика рішення. Для відшукування розв'язку використовується алгоритм сортування, тож асимптотика $O(N \log N)$.

Задача G. Повітряні кульки

N повітряних кульок плавають у повітрі у великій кімнаті Леді після дня народження. Всі кульки плавають в одній площині. Леді любить гратися луком і практикувати свої здібності до полювання. Вона випускає стрілу з лука в площині, у якій плавають кульки, праворуч від лівої сторони кімнати з будь-якої висоти, яку вона обирає. Стріла рухається зліва направо на обраній висоті H , поки не знайде повітряну кульку. У момент, коли стріла торкається повітряної кульки, повітряна кулька лопає

і зникає, а стріла продовжує шлях зліва направо на висоті, яка зменшується на 1. Отже, якщо стріла рухалася на висоті H , після того, як вона потрапила у повітряну кульку, вона рухається на висоті $H-1$.

Мета Леді полягає в тому, щоб вибити всі кульки за допомогою мінімальної кількості пострілів.

Формат вхідних даних

В першому рядку міститься ціле число N ($1 \leq N \leq 1\,000\,000$) – кількість повітряних кульок.

В другому рядку міститься масив N цілих чисел H_i . Кожне ціле число H_i ($1 \leq H_i \leq 1\,000\,000$) – це висота, на якій плаває i -а кулька. Кожна наступна кулька плаває правіше від попередньої.

Формат вихідних даних

Вивести необхідну мінімальну кількість пострілів, яку потрібно зробити Леді, щоб вибити всі повітряні кульки.

Пояснення

Пояснення першого прикладу: Леді стріляє на висоті 5, і тоді знищує кульки з висотами [5, 4, 3], а потім стріляє на висоті 2, – і тоді вибиває кульки з висотами [2, 1].

Приклад вхідних та вихідних даних

Введення	Виведення
5 2 1 5 4 3	2
5 1 2 3 4 5	5
5 4 5 2 1 4	3

Аналіз. Створимо масив H , де $H[k]$ – кількість стріл на висоті k . Тепер якщо розглядаємо чергову кульку і $H[k]$ для висоти цієї кульки більше 0, це значить, що стріла, яку колись випустили, зможе її збити. Тоді треба зменшити $H[k]$ та збільшити $H[k-1]$, де $k-1$ – висота кульки, адже після зіткнення висота стріли зменшиться на 1. Якщо ж $H[k]=0$, то доведеться пустити точно ще одну стрілу, тож треба збільшити відповідь на 1, та збільшити $H[k-1]$, бо після зіткнення ця стріла також знизиться на 1 вниз.

Асимптотика розв'язку. Розв'язок досягається за один прохід через масив, тобто за $O(N)$.

Висновки. Всеукраїнський турнір юних інформатиків – важливий захід для зацікавлення учнів сучасними ІТ-технологіями, профорієнтаційної роботи у цьому напрямку, пошуку та стимулювання обдарованих учнів, формування навичок колективної роботи, компетентностей щодо моделювання, написання алгоритмів та їх кодування тощо. В останні роки спостерігається зростання популярності цього заходу. Серед проблем можна виокремити відсутність онлайн-ресурсу турніру. Сторінка в соціальній мережі Facebook [4] містить тільки епізодичні дані, що стосуються переважно тільки поточного турніру. Тому бажано організувати відповідний сайт для зберігання даних про турніри – умови задач, як першого, так і третього турів, учнівські розв'язки задач першого туру, підсумкові таблиці з результатами тощо.

Список використаних джерел

1. Горошко Ю.В., Мельник В.І., Боркач Є. Система підготовки до олімпіад з інформатики / Ю. Горошко, В. Мельник, Є. Боркач // Science and Education a New Dimension. Pedagogy and Psychology, III (32), Issue: 63, 2015. с. 23-26.

2. Мельник В.І., Горошко Ю.В., Міца О.В. Огляд систем підготовки до олімпіад з інформатики в деяких країнах / В. Мельник, Ю. Горошко, О. Міца // II Всеукраїнської науково-практичної конференції з міжнародною участю «Сучасні інформаційні технології в освіті і науці». – Житомир. – 2017. – С. 21-23.

3. Чернівці – столиця Всеукраїнського турніру з інформатики! [Електронний ресурс]. – Режим доступу: <http://oblosvita.com/navigaciya/skrynka/informatyka/>

4. Всеукраїнський турнір юних інформатиків [Електронний ресурс]. – Режим доступу: <https://www.facebook.com/TUI.Info/>

О ТУРНИРАХ ЮНЫХ ИНФОРМАТИКОВ

Ю.В. Горошко, В.И. Мельник, А.В. Мица

Аннотация. В статье проанализированы Всеукраинские турниры юных информатиков за последние годы. Рассмотрены особенности проведения каждого из трёх туров турнира. Показана позитивная динамика роста количества команд с 2013 года. Предложен разбор задач третьего тура турнира 2017 г. Выяснено, что одной из важных проблем турнира является отсутствие официального онлайн-ресурса турнира, поэтому предлагается \ организовать соответствующий сайт для сохранения данных о турнирах – условий задач как

першого, так и третьего туров, ученических решений задач первого тура, итоговых таблиц с результатами и т.д.

Ключевые слова: турнир юных информатиков, компьютерная программа, моделирование, АСМ.

ABOUT TOURNAMENT OF YOUNG COMPUTER SCIENTISTS

Yu.V. Horoshko, V.I. Melnyk, A.V. Mitsa

Resume. All-Ukrainian tournaments of young computer scientists in recent years have been analyzed in the article. The peculiarities of each of the three rounds of the tournament are considered. The positive dynamics of the growth of the number of teams since 2013 is shown. The analysis of tasks of the third round of the tournament of 2017 is offered. It was found out that one of the important problems of the tournament is the lack of an official online tournament resource, therefore it is proposed to organize an appropriate site for storing data about tournaments – the conditions for the tasks of both the first and third rounds, the student's solutions of the tasks of the first round, the final tables with results etc.

Keywords: tournaments of young computer scientists, computer program, modeling, АСМ.

DOI 10.31392/NPU-nc.series2.2018.20(27).06

УДК 378.018.43:004.946.5

І.С. Войтович¹, Ю.С. Трофименко²

¹доктор педагогічних наук, професор

²аспірант кафедри комп'ютерної інженерії та освітніх вимірювань

Національний педагогічний університет імені М.П. Драгоманова

ОСОБЛИВОСТІ ВИКОРИСТАННЯ GOOGLE CLASSROOM ДЛЯ ОРГАНІЗАЦІЇ ДИСТАНЦІЙНОГО НАВЧАННЯ СТУДЕНТІВ

Анотація. Стаття присвячена застосуванню сучасних інформаційних технологій в освіті. Висвітлено особливості роботи з системою Google Classroom у дистанційному навчанні.

Ключові слова: дистанційне навчання, віртуальний клас, мобільність навчання, Google Classroom.

Протягом останніх двох десятиріч років, відбувається процес інтенсивного впровадження в систему традиційного навчання на базі сучасних інформаційно-комунікаційних технологій та їх гармонійного поєднання. Це стало можливим здебільшого з розвитком мережі Інтернет, що дало можливість передавати необхідну кількість даних з одного кінця світу в інший, вільно вести дискусії з іншими користувачами мережі в online режимі і розміщувати необхідні повідомлення на Інтернет-сайтах, роблячи їх доступними для всіх бажаючих.

Проблеми застосування дистанційного навчання у навчальному процесі не є новими. До проблем застосування дистанційного навчання у навчальному процесі звертались багато вчених і практиків, зокрема подібні проблеми є предметом досліджень В.В. Дивака, Н.О. Думанського, Т.Г. Крамаренко, В.В. Попова, В.С. Трохименка та ін. Сутність дистанційного навчання й особливості його використання у процесі професійної підготовки фахівців у ВНЗ розглянуто у роботах В.Ю. Бикова, А.В. Хуторського, В.М. Кухаренко, Н.Г. Сиротенко, О.В. Рибалки.

Висвітлення проблем, пов'язаних з використанням сучасних інформаційних та комп'ютерних технологій у педагогічному процесі, започатковано і розвинуто в фундаментальних роботах учених (В.Ю. Бикова, Р. Вільямса, Б.С. Гершунського, В.М. Глушкова, А.Н. Єршова, М.І. Жалдака, С.Г. Литвинової, Ю.І. Машбиця, Н.В. Морзе, С. Пейперта, Є.С. Полата, М.П. Шишкіної та ін.). У роботах цих авторів показано, що впровадження комп'ютерних технологій у практику навчання є однією з форм підвищення ефективності педагогічного процесу.

Галузь освіти в цілому і окремі освітні процеси потребують інформатизації, тому що, по-перше, підсилення інформаційними технологіями освітнього процесу підвищує його ефективність і привабливість; по-друге, учні і студенти, викладачі і вчителі повинні вміти використовувати інформаційні технології [8]. Крім того, сучасні інформаційні технології можна класифікувати як презентаційні, постачання та взаємодії.

Презентаційні технології включають: книги та друковані матеріали; електронні тексти та публікації; комп'ютерні навчальні програми; мультимедіа; телебачення; радіо; віртуальну реальність та моделювання; електронні підтримувальні системи [2].

Технології постачання (синхронні та асинхронні) включають такі: радіотрансляція; аудіокасети; телетрансляція; відеокасети; CD-ROM; DVD (цифрові відеодиски); Інтернет, Інтранет.

Технології взаємодії включають: телеконференції; електронну пошту; групову мережу.

В умовах величезного потоку даних і дефіциту навчального часу ведеться активний пошук нових резервів для створення гнучкої і мобільної системи навчання. У зв'язку зі змінами у навчальних