

Волинська учнівська інтернет-олімпіада з інформатики



Подано особливості організації та проведення Волинської учнівської інтернет-олімпіади з інформатики, приклади розв'язування задач мовою «Пайтон».

Ключові слова: олімпіада, інформатика, «Ідждж», пайтон.

His I. V. Volyn Student Internet Olympiad in Informatics.

Features of the organization and conduct of the Volyn Student Internet Olympiad in Informatics and examples of solving problems in the Python language are presented.

Keywords: Olympiad, computer science, Ejudge, Python.

Постановка питання. З метою розвитку освітнього, інноваційного, інформаційного потенціалу Волинської області, упровадження нових форм і методів пошуку обдарованих учнів та створення умов для розвитку їх здібностей із 2004 року проводиться Волинська учнівська інтернет-олімпіада з інформатики.

Виклад основного матеріалу. Це щорічне регіональне інтелектуальне змагання серед учнів 8–11 класів закладів загальної середньої освіти області для стимулювання творчого самовдосконалення школярів, інтересу до вивчення алгоритмізації та програмування, виявлення талановитих учнів, які мають потенціал для подальшого розвитку в галузі програмування.

Офіційним сайтом проведення олімпіади є сайт Волинського інституту післядипломної педагогічної освіти [1]. Він забезпечує:

- розміщення офіційних наказів щодо проведення і підсумків олімпіади;
- реєстрацію її учасників з автоматичним розсиленням кодів і паролів учасникам;
- розміщення завдань турів олімпіади з посиланням на систему автоматичного тестування розв'язків;
- висвітлення результатів учасників по кожному туру і загального рейтингу;
- умови завдань, тести та авторські розв'язання.

Олімпіада проводиться у шість турів. Тривалість виконання завдань кожного туру – один тиждень.

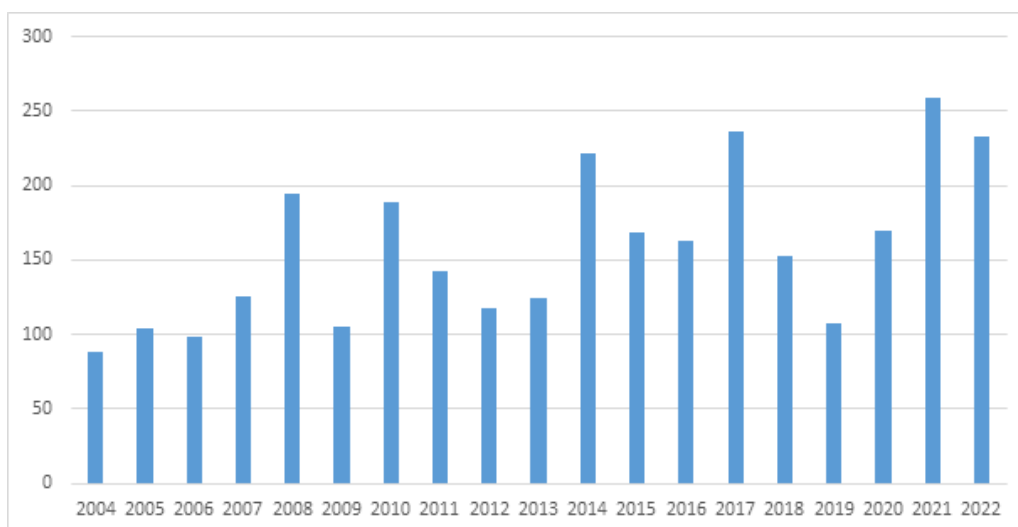


Рисунок 1. Кількість учасників олімпіади

Для автоматичного тестування та оцінювання рішень використовується система «Іджадж» (Ejudge) [2].

Іджадж дозволяє створювати й організувати олімпіаду з інформатики. Система має багато корисних функцій для забезпечення безпеки, цілісності та конфіденційності результатів тестів. Іджадж підтримує багато мов програмування, включаючи C++, «Джава» (Java), «Пайтон» (Python), «Паскаль» та інші. Кожне рішення учасника олімпіади перевіряється за допомогою різних тестів, що дозволяє оцінити правильність та оптимальність виконання завдань.

Система має набір корисних функцій для організаторів олімпіади:

- установлює терміни виконання завдань;
- додає коментарі до рішень;
- організовує обговорення завдання;
- зберігає історію тестування.

Іджадж є потужною системою для автоматизованої перевірки програмних завдань, яка допомагає якісно провести олімпіаду з інформатики.

Завдання можуть бути різної складності, від базових до складних, що вимагають глибокого розуміння теорії алгоритмів та досвіду розв'язування складних задач. Завдання можуть бути з різних галузей інформатики, таких як теорія алгоритмів, структури даних. Можуть включати в себе виконання програм, аналіз даних, визначення правильності алгоритмів, розв'язання математичних задач та інші.

Тематика завдань олімпіади з інформатики досить широка й охоплює різні аспекти програмування, алгоритмів та структур даних. Основні теми, які можуть бути охоплені, включають:

– алгоритми та структури даних (це може включати роботу з різними типами даних, такими як масиви, списки, дерева, графи, а також алгоритми сортування та пошуку);

– синтаксичний і лексичний розбір виразів; процес перетворення виразу на лексеми, що можна використовувати для подальшого аналізу та обчислень. Вони також використовуються в математичних системах та інших програмах для обробки виразів і формул;

– обчислювальну геометрію (вивчає математичні алгоритми й методи розв'язання задач, пов'язаних із геометричними об'єктами та їх взаємодією. До таких задач можна віднести пошук перетинів, визначення відстаней між об'єктами, визначення зон доступу, рух по мапі, оптимізацію траєкторій тощо);

– жадібні алгоритми (це підхід до розв'язання оптимізаційних задач, коли на кожному кроці алгоритму вибирається найбільш оптимальне рішення, доступне на даний момент, для досягнення загальної оптимальності);

– динамічне програмування (це ефективний метод для розв'язання таких задач, де потрібно знайти

оптимальне рішення з великої кількості можливих варіантів);

– теорію графів: вивчає структури, які складаються з вершин (або вузлів) та ребер (або зв'язків) між ними. Графи можуть бути застосовані для візуалізації та аналізу складних систем, таких як мережі зв'язків у комп'ютерних мережах, транспортні маршрути, соціальні мережі тощо.

Авторські рішення задач олімпіади [3]

Задача 1. Відстань Геммінга

Відстань Геммінга (англ. Hamming distance) між двома рядками символів однакової довжини – це кількість позицій, у яких відповідні символи відрізняються.

Наприклад:

Для рядків „karolin” і „kathrin” відстань Геммінга дорівнює 3.

Для рядків „karolin” і „kerstin” відстань Геммінга дорівнює 3.

Для рядків **0000** і **1111** відстань Геммінга дорівнює 4.

Для рядків **2173896** і **2233796** відстань Геммінга дорівнює 3.

Дано два додатних числа n і m у десятковій системі числення ($0 < n < 10^6$, $0 < m < 10^6$). Потрібно обчислити відстань Геммінга між цими двома числами у двійковій системі.

Вхідні дані

Два цілі числа, розділені пропуском.

Вихідні дані

Відстань Геммінга як ціле число.

Приклади вхідних і вихідних даних

Вхідні дані	Вихідні дані
16 16	0
140 160	3
16 128	2

Код розв'язання пайтоном

```
n, m = map(int, input().split())
print(bin(n ^ m).count('1'))
```

Аналіз коду розв'язання

Цей код отримує два числа n та m із вхідного потоку за допомогою функції `input()`, розбиває їх на окремі числа за допомогою функції `map(int, input().split())` та присвоює їх змінним n та m .

Далі виконується операція бітового відображення XOR (`^`) над цими числами, що дає результат, який включає 1 у бінарному вигляді там, де відповідні біти в n та m відрізняються.

Функція `bin()` перетворює цей результат у бінарний вигляд, після чого метод `count()` використовується для підрахунку кількості одиниць у рядку.

Отже, код виводить кількість бітів, у яких два введені числа n та m відрізняються.

Задача 2. «Хюндай»

Микола часто їздить по роботі зі Львова до Києва і назад. Оскільки справи в нього завжди термінові, добирається він швидкісним потягом «Хюндай». Відомо, що в кожному його вагоні є рівно K місць, а саме $K/2$ рядів по два місця в кожному. (K – парне). Якось дорогою додому після ділової зустрічі у Миколи не було сусіда, і йому стало нудно. Тому його зацікавило: скільки найбільше людей можна посадити у цей вагон, щоб рівно в половини був сусід. Допоможіть Миколі дати відповідь на це складне питання.

Вхідні дані

У єдиному рядку вхідного файлу (або стандартного потоку введення) дане натуральне число K ($2 \leq K \leq 1000\ 000\ 000$).

Вихідні дані

У єдиному рядку вихідного файлу виведіть максимальну кількість людей, яких можна посадити так, щоб рівно в половини з них був сусід.

Приклади вхідних і вихідних даних

Вхідні дані	Вихідні дані
20	12

Код розв'язання мовою «Пайтон»

```
print(int(input())// 6 * 4)
```

Аналіз коду розв'язання

Розв'язком задачі буде формула $k = n/2/3*4$.

Пояснимо виведення формули за допомогою рис. 2 при $n = 20$:

1	1
1	1
1	1
2	2
2	2
2	2
3	3
3	3
3	3

Рисунок 2

Нам потрібно розділити половину місць (кількість рядів) на три рівні групи, які на рисунку пронумеровано 1, 2, 3, і взяти 4 (адже в першій групі люди сидять по двоє, в інших двох – по одному). Зайняті місця зафарбовано.

Задача 3. Повінь

Оцінити площу затоплення під час повені, використовуючи схему поперечного перерізу ділянки

ландшафту (див. рис. 3). Вважається, що дощ іде безперервно і затоплює всі низини, а по обидва боки кожної зі схем знаходиться море, куди стікає зайва вода.

Вхідні дані

Ви отримуєте схему поперечного перерізу ділянки ландшафту у вигляді рядка ($1 \leq$ довжина рядка схеми ≤ 500), що складається із символів „/” і „\”, які позначають пологі схили ландшафту, і символу „_”, що позначає рівнини. Наведена на рисунку схема поперечного перерізу описується рядком:

\//\/_\//\//_//__V_V_/

У цьому разі кількість усього затоплених клітинок дорівнює 35. Кількість окремих неперервних затоплених областей дорівнює п'яти – це ділянки з площами 4, 2, 1, 19 і 9 клітинок відповідно.

Вихідні дані

Надрукувати інформацію про зони затоплень, починаючи від лівої межі схеми, у такому форматі (цілі числа):

- в першому рядку – всього затоплених клітинок;
- у другому рядку – кількість окремих неперервних затоплених областей;
- у третьому рядку – через пропуск значення площ затоплених областей.

Якщо ж затоплені клітинки відсутні, то надрукувати 0.

Приклади вхідних і вихідних даних

Вхідні дані	Вихідні дані
\//\/_\//\//_//__V_V_/	35 5 4 2 1 19 9
///_V//_\\	11 1 11
\/__	0

Код розв'язання мовою «Пайтон»

```
diagram = input()
slopes, floods = [], []

for i, ch in enumerate(diagram):
    if ch == „\”:
        slopes.append(i)
    elif ch == „/” and slopes:
        k = slopes.pop()
        flood = i - k
        floods.append((k, flood))
    while len(floods) > 1 and k < floods[-2][0]:
        floods.append((k, floods.pop()[1] + floods.pop()[1]))
```

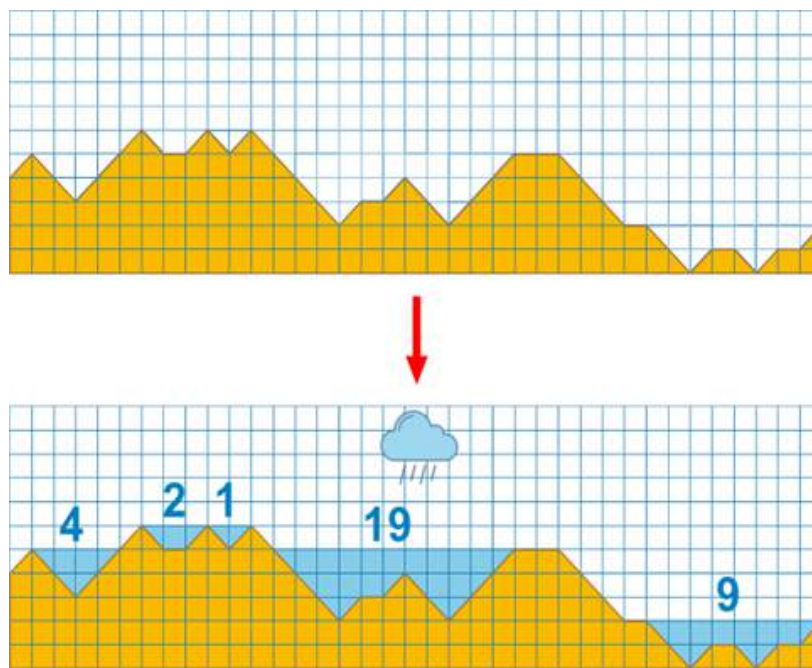


Рисунок 3

```
r = [flood for _, flood in floods]
if r:
    print(sum(r))
    print(len(r))
    print(' '.join(map(str, r)))
else:
    print(0)
```

Аналіз коду розв'язання

Ідея розв'язку полягає у реалізації алгоритму справжнього розливу води. Послідовно заносимо карту, фіксуємо висоту. Знайшовши, шукаємо пару, куди дійде вода, рахуємо об'єм води і заносимо початок і кінець рівня.

В таблиці floods аналізуємо рівні й шукаємо спільний рівень.

Цей код приймає рядок, який подає пейзаж із пагорбами та долинами, де кожен символ у рядку становить одиницю ширини. Символи / та \ позначають схили, що йдуть угору та вниз відповідно. Код обчислює кількість води, яка накопичилася б у долинах між пагорбами, якби пішов дощ, і друкує загальну кількість води, кількість долин, які містять воду, і кількість води в кожній долині.

Алгоритм працює, відстежуючи індекси всіх схилів, що йдуть униз (тобто індекси всіх символів \). Коли він зустрічає схил, що йде вгору (тобто символ /), то витягує зі списку схилів індекс останнього схилу, що йде вниз, обчислює ширину долини між двома схилами та додає цю інформацію як кортеж до списку повеней. Якщо між однією парою схилів є кілька долин різної ширини, вони об'єднуються в одну долину.

Нарешті, код витягує ширину всіх долин зі списку повеней і виводить загальну кількість води, кількість долин з водою та ширину кожної долини.

Висновки. Волинська учнівська інтернет-олімпіада в основному спрямована на розвиток навичок алгоритмічного мислення, програмування та підготовки до Всеукраїнської олімпіади з інформатики. Учасники мають демонструвати свої знання та вміння при розв'язуванні різних задач, котрі вимагають створення ефективних алгоритмів. Олімпіада дає змогу учням проявити свої таланти в галузі програмування, а також підготуватися до роботи у сфері інформаційних технологій. Переможці змагань мають можливість взяти участь у III етапі Всеукраїнської учнівської олімпіади з інформатики.

Література

1. Волинська учнівська Інтернет-олімпіада з інформатики. URL: <http://176.102.48.88/vipoolimp> (дата звернення: 08.03.2023).
2. Інструкція по користуванню системою Ejudge. URL: <http://176.102.48.88/vipoolimp/joomla/images/stories/2021-2022/ejudge-quide.pdf> (дата звернення: 08.03.2023).
3. Умови завдань, тести, авторські розв'язки олімпіади. *Волинська учнівська Інтернет-олімпіада у 2022 році*. URL: https://drive.google.com/drive/folders/16kG9Bknk7ps_hkGZ_nLVeffE9eK2uuxf (дата звернення: 08.03.2023).