

Київський національний університет імені Тараса Шевченка

КРЕНЕВИЧ А.П.

**Методичні вказівки
до лабораторних занять із дисципліни
«Алгоритми і структури даних»**

для студентів механіко-математичного факультету

Київ – 2018

УДК 519.942+550

Рецензенти:
доктор фіз.-мат. наук, професор
доктор фіз.-мат. наук,

*Рекомендовано до друку вченою радою механіко-математичного
факультету
(протокол № __ від __ _____ 201__ року)*

Крєневич А.П.

Методичні вказівки до лабораторних занять із дисципліни «Алгоритми і структури даних» для студентів механіко-математичного факультету – К.: ВПЦ "Київський Університет", 2018. – __ с.

Посібник містить перелік завдань для самостійної роботи з дисципліни «Алгоритми і структури даних», що викладається студентам механіко-математичного факультету Київського національного університету імені Тараса Шевченка. Він містить завдання для засвоєння основних понять цього курсу, таких як рекурентні співвідношення, рекурсія, аналіз складності алгоритмів, лінійні рекурсивні структури даних, графи, дерева, динамічне програмування тощо.

Для студентів механіко-математичного факультету та викладачів, які проводять заняття з курсу «Алгоритми і структури даних».

ЗМІСТ

ВСТУП	4
ВИМОГИ ДО ОФОРМЛЕННЯ ЛАБОРАТОРНИХ РОБІТ	5
ЛАБОРАТОРНА РОБОТА 1. СКЛАДНІСТЬ АЛГОРИТМІВ	7
ЛАБОРАТОРНА РОБОТА 2. ЛІНІЙНИЙ ТА БІНАРНИЙ ПОШУК.....	13
ЛАБОРАТОРНА РОБОТА 3. РОЗВ'ЯЗАННЯ РІВНЯНЬ МЕТОДОМ БІЕКЦІЇ.	16
ЛАБОРАТОРНА РОБОТА 4. БІНАРНИЙ ПОШУК ПО ВІДПОВІДІ.	18
ЛАБОРАТОРНА РОБОТА 5. ХЕШУВАННЯ ТА ХЕШ-ТАБЛИЦІ.....	21
ЛАБОРАТОРНА РОБОТА 6. СОРТУВАННЯ.....	23
ЛАБОРАТОРНА РОБОТА 7. РЕКУРСІЯ, ПОВНИЙ ПЕРЕБІР, МЕТОД «РОЗДІЛЯЙ І ВОЛОДАРЮЙ».....	25
ЛАБОРАТОРНА РОБОТА 8. СТЕК ТА ЙОГО ЗАСТОСУВАННЯ.....	27
ЛАБОРАТОРНА РОБОТА 9. ЧЕРГИ ТА ЗВ'ЯЗНІ СПИСКИ. ЇХНЕ ЗАСТОСУВАННЯ	29
ЛАБОРАТОРНА РОБОТА 10. ДЕРЕВА. АЛГОРИТМИ НА ДЕРЕВАХ.	30
ЛАБОРАТОРНА РОБОТА 11. БІНАРНІ ДЕРЕВА. БІНАРНІ ДЕРЕВА ПОШУКУ.....	31
ЛАБОРАТОРНА РОБОТА 12. БІНАРНА КУПА ТА ДЕРЕВО ВІДРІЗКІВ.	32
ЛАБОРАТОРНА РОБОТА 13. АЛГОРИТМИ НА НЕ ЗВАЖЕНИХ ГРАФАХ.	33
ЛАБОРАТОРНА РОБОТА 14. АЛГОРИТМИ НА ЗВАЖЕНИХ ГРАФАХ	35
ЛАБОРАТОРНА РОБОТА 15. ЛАБІРИНТИ.....	36
СПИСОК ЛІТЕРАТУРИ ТА ДОДАТКОВИХ ДЖЕРЕЛ	38

ВСТУП

ВИМОГИ ДО ОФОРМЛЕННЯ ЛАБОРАТОРНИХ РОБІТ

1. Основна ціль лабораторної роботи довести викладачу, що студент засвоїв матеріал на достатньому рівні.
2. Кожна лабораторна роботи містить
 - a. перелік контрольних запитань, що стосуються теоретичного матеріалу;
 - b. варіанти індивідуальних/групових завдань лабораторних робіт для практичного виконання;
 - c. посилання на допоміжну літературу для закріплення теоретичного матеріалу і стане в нагоді під час виконання практичних завдань.
3. На вибір студентів надано перелік практичних завдань по кожній темі, серед яких вони мають можливість самостійно вибрати бажаний варіант. Обраний номер завдання студенти записують у спеціальний хмарний документ, наданий викладачем, ціль якого рівномірно розподілити завдання між студентами групи. Кілька студентів однієї групи, що можуть обрати один варіант не може перевищувати 2, у випадку самостійної роботи, та 3-х, у випадку роботи у команді.
4. Кожна лабораторна робота оформлюється у паперовому та електронному вигляді.
5. Паперове оформлення здійснюється за допомогою текстового процесору Word (чи аналогічного) або редактора презентацій PowerPoint (чи аналогічного) та має містити такі частини:
 - a. Титульний аркуш, що містить назву навчальної дисципліни, номер лабораторної роботи, номер варіанту, ПІБ студента (студентів, якщо спільна лабораторна робота), що виконав роботу, спеціальність, курс, номер групи.
 - b. Умову задачі. Якщо задача береться з електронних джерел, то вказати посилання на задачу. Умова задачі має містити приклади вхідних та вихідних даних.
 - c. Аналіз задачі та опис алгоритму, яким пропонується розв'язувати задачу.
 - d. Приклад роботи алгоритму на модельному прикладі (взятому з умови задачі або придуманому самостійно).
 - e. Програмну реалізацію (основні моменти).
 - f. Висновки: якщо робота програми перевірялася за допомогою електронних систем, то обов'язково вказати

посилання на систему перевірки разом з задачею (наприклад, <https://www.e-olymp.com/uk/problems/3966>), логін (логіни) виконавців лабораторної роботи та результат перевірки (наприклад, 100%). Якщо результат не 100%, то обов'язково зазначити у чому проблема (не пройшов, по часу, помилки виконання, тощо).

6. Електронна частина, складається з
 - a. Файлу (файлів), з яких друкується паперовий примірник лабораторної роботи.
 - b. Файлів вихідного коду програми, що розв'язує поставлену задачу.
 - c. Файлів вхідних даних (за необхідності) – текстові файли, що містять вхідні дані задачі. Вони мають бути названі згідно з правилом – «input.txt» (якщо один файл) та «input01.txt», «input02.txt» і т.д. – якщо кілька файлів вхідних даних.
7. Електронний варіант надсилається на електронну адресу викладача, що викладає практичні заняття не пізніше ніж напередодні лабораторного заняття, на якому студент має її захищати.
8. Захист лабораторної роботи здійснюється студентом (студентами) одним з двох способів:
 - a. Індивідуальна співбесіда з викладачем, протягом якої викладач ставить запитання по теоретичному матеріалу теми лабораторної роботи (з переліку контрольних запитань), вислуховує ідею алгоритму, що розв'язує поставлену задачу, знайомиться з деталями реалізації алгоритму та перевіряє коректність розв'язку.
 - b. Презентація розв'язаної задачі з усіма деталями реалізації для всіх студентів навчальної групи. У цьому випадку роздрукований варіант не вимагається.

ЛАБОРАТОРНА РОБОТА 1. Складність алгоритмів

Контрольні запитання

1.1.

Варіанти індивідуальних завдань

1.1. Визначте час виконання фрагментів програм заданих нижче

a)

```
1 i = 0
2 while i < n:
3     k += 1
4     i += 1
```

b)

```
1 i = 1
2 while i < n:
3     k += 1
4     i = i * 2
```

c)

```
1 i = n - 1
2 while i != 0:
3     k += 1
4     i = i / 2
```

d)

```
1 i = 0
2 while i < n:
3     if i % 2 == 0:
4         k += 1
5     i += 1
```

e)

```
1 i = 0
2 while i < n:
3     j = 0
4     while j < n:
5         k += 1
6         j += 1
7     i += 1
```

f)

```
1 i = 0
2 while i < n:
3     j = i
4     while j < n:
5         k += 1
6         j += 1
7     i += 1
```

g)

```
1 i = 0
2 while i < n:
3     j = 0
4     while j < i * i:
5         k += 1
6         j += 1
7     i += 1
```

h)

```
1 i = 0
2 while i < n:
3     j = n
4     while j != 0:
5         k += 1
6         j /= 3
7     i += 1
```

1.2. Визначте час виконання програми у явному вигляді, якщо для нього відоме рекурентне співвідношення

- a) $T(n) = \begin{cases} 1, & n = 0; \\ T(n-1) + 1, & n \geq 1. \end{cases}$
- b) $T(n) = \begin{cases} 1, & n \leq a, a > 0; \\ T(n-a) + 1, & n > a. \end{cases}$

$$\text{c) } T(n) = \begin{cases} 1, & n = 0; \\ 2T(n-1) + 1, & n > 1. \end{cases}$$

$$\text{d) } T(n) = \begin{cases} 1, & n = 0; \\ 2T(n-1) + n, & n > 1. \end{cases}$$

$$\text{e) } T(n) = \begin{cases} 1, & n = 1; \\ T(n/2) + 1, & n \geq 2. \end{cases}$$

$$\text{f) } T(n) = \begin{cases} 1, & n = 1; \\ 2T(n/2) + 1, & n \geq 2. \end{cases}$$

$$\text{g) } T(n) = \begin{cases} 1, & n = 1; \\ 2T(n/2) + n, & n \geq 2. \end{cases}$$

1.3. Нехай $f(n) = 3n^2 - n + 4$. Користуючись означенням покажіть що

$$\text{a) } f(n) = O(n^2), \quad \text{b) } f(n) = \Omega(n^2).$$

1.4. Нехай $f(n) = 3n^2 - n + 4$ та $g(n) = n \log n + 5$. Покажіть що

$$f(n) + g(n) = O(n^2).$$

1.5. Нехай $f(n) = \sqrt{n}$ та $g(n) = \log n$. Покажіть що

$$f(n) + g(n) = O(\sqrt{n}).$$

1.6. Для кожної пари функцій $f(n)$ та $g(n)$, зазначених у таблиці визначте яке із співвідношень має місце $f(n) = O(g(n))$ чи $g(n) = O(f(n))$

$f(n)$	$g(n)$
$10n$	$n^2 - 10n$
n^3	$n^2 \log n$
$n \log n$	$n + \log n$
$\log n$	$\sqrt[k]{n}$
$\ln n$	$\log n$
$\log(n+1)$	$\log n$
$\log \log n$	$\log n$
2^n	10^n
n^m	m^n
$\cos(n\pi/2)$	$\sin(n\pi/2)$
n^2	$(n \cos n)^2$

1.7. Знайдіть асимптотичний час виконання програми у явному вигляді, якщо для нього відоме рекурентне співвідношення

$$\text{a) } T(n) = \begin{cases} O(1), & n = 0; \\ aT(n-1) + O(1), & n \geq 1, a > 1. \end{cases}$$

$$\text{b) } T(n) = \begin{cases} O(1), & n = 0; \\ aT(n-1) + O(n), & n \geq 1, a > 1. \end{cases}$$

- c) $T(n) = \begin{cases} O(1), & n = 0; \\ aT([n/a]) + O(1), & n \geq 1, a \geq 2. \end{cases}$
- d) $T(n) = \begin{cases} O(1), & n = 0; \\ aT([n/a]) + O(n), & n \geq 1, a \geq 2. \end{cases}$

1.8. Доведіть співвідношення

$$\begin{array}{lll} \text{a)} \quad \sum_{i=0}^n i = O(n) & \text{b)} \quad \sum_{i=0}^n i^2 = O(n^2) & \text{c)} \quad \sum_{i=0}^n i^3 = O(n^4) \\ \text{d)} \quad \sum_{i=0}^n a^i = O(n) & \text{e)} \quad \prod_{i=1}^n \frac{1}{1+i} = O(n) & \text{f)} \quad \prod_{i=1}^n \frac{1}{1+i^2} = O(n^2) \\ \text{g)} \quad \prod_{i=1}^n \frac{1}{1+i!} = O(n) & \text{h)} \quad \prod_{i=1}^n \frac{1}{1+i^m} = O(nm) & \text{i)} \quad \prod_{i=1}^n \frac{1}{1+i^i} = O(n^2) \end{array}$$

1.9. Чи можна описати алгоритм для кожної з задач підрахунку суми наведених нижче, асимптотична складність яких буде $O(1)$? Якщо так, то наведіть фрагмент такої програми.

$$\begin{array}{lll} \text{a)} \quad \sum_{i=0}^n i & \text{b)} \quad \sum_{i=0}^n a^i & \text{c)} \quad \sum_{i=0}^{\infty} a^i, |a| \leq 1 \end{array}$$

1.10. Чи можна описати алгоритм для кожної з задач підрахунку суми наведених нижче, асимптотична складність яких буде $O(n)$? Якщо так, то наведіть фрагмент такої програми.

a) $\sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}, (n \text{ коренів});$

b) $y = x^{2^n} + x^{2^{n-1}} + \dots + x^4 + x^2 + 1;$

c) $\left(1 + \frac{1}{1^1}\right) \left(1 + \frac{1}{2^2}\right) \dots \left(1 + \frac{1}{n^n}\right);$

d) $1 + \sin x + \dots + \sin^n x.$

1.11. Припустимо, що n, m та k невід'ємні цілі числа і методи e, f, g та h мають такі характеристики:

- Час виконання у найгіршому випадку методу e $e(n, m, k) \in O(1)$ і повертає значення з проміжку від 1 до $(n + m + k)$.
- Час виконання у найгіршому випадку методу f $f(n, m, k) \in O(n + m)$.

- Час виконання у найгіршому випадку методу $g(n, m, k)$ є $O(m + k)$.
- Час виконання у найгіршому випадку методу $h(n, m, k)$ є $O(n + k)$.

Визначте асимптотичні оцінки виконання програм у найгіршому випадку в термінах O —«великого» для фрагментів програм зазначених нижче:

- a)

```
1 f(n, 10, 0)
2 g(n, m, k)
3 h(n, m, 1000000)
```
- b)

```
1 for i in range(e(n, 10, 100)):
2     f(n, 10, 0);
```
- c)

```
1 for i in range(n):
2     f(n, m, k);
```
- d)

```
1 for i in range(n):
2     for j in range(i, n):
3         f(n, m, k);
```

1.12. Визначте асимптотичну оцінку виконання функції у найгіршому випадку в термінах O —«великого» для функції:

```
def f(n):
    sum = 0
    for i in range(1, n + 1):
        sum = sum + i
    return sum
```

Що є результатом виконання наведеної функції для заданого натурального числа n ? Чи можна оптимізувати цю функцію, покращивши її асимптотичну оцінку?

1.13. Нехай $f(n)$ функція визначена у праві 1.12. Розглянемо функцію

```
def g(n):
    sum = 0
    for i in range(1, n + 1):
        sum = sum + i + f(i)
    return sum
```

Визначте асимптотичну оцінку виконання функції $g(n)$ у найгіршому випадку в термінах O —«великого». Що є результатом виконання функції $g(n)$ для

заданого натурального числа n . Чи можна оптимізувати цю функцію, покращивши її асимптотичну оцінку?

1.14. Нехай $f(n)$ функція визначена у вправі 1.12, а функція $g(n)$ – у вправі 1.13. Розглянемо функцію

```
def h(n):  
    return f(n) + g(n)
```

Визначте асимптотичну оцінку виконання функції $h(n)$ у найгіршому випадку в термінах O –«великого». Що є результатом її виконання для заданого натурального числа n . Чи можна оптимізувати цю функцію, покращивши її асимптотичну оцінку?

1.15. Визначте асимптотичну оцінку виконання функції у найгіршому випадку в термінах O –«великого» для функції:

```
def f(n):  
    k = 0  
    i = n - 1  
    while i != 0:  
        k += 1.0 / i  
        i = i / 2  
    return k
```

Що є результатом виконання наведеної функції для заданого натурального числа n ?

1.16. Опишіть функції натурального аргументу n , час виконання яких у найгіршому випадку має асимптотику:

- | | | |
|----------------|------------------|-------------|
| a) $O(n)$ | b) $O(n^2)$ | c) $O(n^3)$ |
| d) $O(\log n)$ | e) $O(n \log n)$ | f) $O(2^n)$ |

1.17. Опишіть функції натуральних аргументів n та m , час виконання яких у найгіршому випадку має асимптотику:

- | | | |
|---------------|------------------|-------------|
| a) $O(n + m)$ | b) $O(m \log n)$ | c) $O(n^m)$ |
|---------------|------------------|-------------|

1.18. Натуральне число називається паліндромом, якщо його запис читається однаково зліва направо і справа наліво (наприклад, 1, 393, 4884). Скласти програму, що визначає, чи є задане натуральне число n паліндромом асимптотична складність якої $O(n)$, де n – кількість цифр у числі.

1.19. Числами трибоначчі називається числова послідовність $\{T_k: k \geq 0\}$, задана рекурентним співвідношенням третього порядку:

$$T_0 = 0, T_1 = T_2 = 1, T_k = T_{k-1} + T_{k-2} + T_{k-3}, \quad k \geq 3.$$

Опишіть функції для обчислення T_n за допомогою рекурентного співвідношення та використовуючи рекурсію. Обчисліть асимптотичну

складність кожного з варіантів. Порівняйте абсолютний час виконання (у секундах) обох варіантів для знаходження T_{10}, T_{20}, T_{50} .

1.20. Послідовністю Падована називається числова послідовність $\{P_k: k \geq 0\}$, задана рекурентним співвідношенням третього порядку:

$$P_0 = P_1 = P_2 = 1, P_k = P_{k-1} + P_{k-3}, \quad k \geq 3.$$

Опишіть функції для обчислення P_n за допомогою рекурентного співвідношення та використовуючи рекурсію. Обчисліть асимптотичну складність кожного з варіантів. Порівняйте абсолютний час виконання (у секундах) обох варіантів для знаходження P_{10}, P_{20}, P_{50} .

ЛАБОРАТОРНА РОБОТА 2. ЛІНІЙНИЙ ТА БІНАРНИЙ ПОШУК.

Контрольні запитання

- 2.1. Які задачі допомагає розв'язати лінійний пошук? Наведіть алгоритм лінійного пошуку та його складність (у найгіршому випадку).
- 2.2. У чому полягає цілочисельний бінарний пошук? Для яких структур даних він може бути реалізований? Яка складність бінарного пошуку? Яка перевага бінарного пошуку у порівнянні з лінійним пошуком? Наведіть алгоритм бінарного пошуку.

Завдання для аудиторної роботи

- 2.1. Одиниці

<https://www.e-olymp.com/uk/problems/622>

На уроках інформатики вас, напевно, вчили переводити числа з одних систем числення у інші і виконувати інші подібні операції. Прийшов час продемонструвати ці знання. Знайдіть кількість одиниць у двійковому запису заданого числа.

Вхідні дані

У вхідному файлі міститься єдине ціле число n ($0 \leq n \leq 2000000000$).

Вихідні дані

Вихідний файл повинен містити одне число — кількість двійкових одиниць у запису числа n .

Вхідні дані #1

5

Вихідні дані #1

2

Вхідні дані #2

7

Вихідні дані #2

3

Зауваження. Скористайтеся побітовими операціями.

- 2.2. Реалізуйте інтерфейс для роботи з англійсько-українським словником та швидким пошуком перекладу. Реалізацію здійсніть у вигляді сукупності двох функцій описаних нижче. Функція

```
def addTranslation(eng, translation)
```

додає до словника англійське слово `eng` та його переклад `translation`. Пари `(eng, translation)` приходяться у порядку, що

відповідає лексикографічному порядку. Функція

```
def find(eng)
```

повертає переклад слова `eng` зі словника, якщо воно міститься у словнику, або порожній рядок у іншому разі.

Тестова програма `main.py`, файл `user.py`, що містить вищезгаданий інтерфейс та вхідні дані (текстові файли) розташовані за посиланням:

<https://github.com/krenevych/algo/tree/master/labs/T3/Search/task1>

Завантажте всі файли, що містяться за цим посиланням, у одну папку та реалізуйте згаданий вище інтерфейс (файл `user.py`). Для перевірки правильності алгоритму запустіть файл `main.py`.

- 2.3. Реалізуйте алгоритм пошуку номеру найпершого входження до заданого масиву, заданого числа `x`. Якщо заданий елемент відсутній у списку - поверніть номер першого елементу, що більший за число `x`:

```
array[i] >= x
```

Алгоритм оформіть у вигляді функції

```
def bsearch_leftmost(array, x)
```

Гарантується, що вхідний масив `array` впорядкований за неспаданням.

Тестова програма розташована за посиланням:

<https://github.com/krenevych/algo/tree/master/labs/T3/Search/task2>

Завантажте всі файли, що містяться за цим посиланням, у одну папку. Реалізуйте згадану вище функцію, інтерфейс якої міститься у файлі `user.py`. Для перевірки правильності алгоритму запустіть файл `main.py`.

Завдання для самостійної роботи

- 2.4. Циклічні зсуви
<https://www.e-olymp.com/uk/problems/27>
- 2.5. Затятий колекціонер метеликів
<https://www.e-olymp.com/uk/problems/3966>
- 2.6. Реалізуйте алгоритм пошуку номеру останнього входження до заданого масиву заданого числа `x`. Якщо такий елемент відсутній у списку - поверніть номер останнього елементу, що менший за число `x`. Зауважимо, що якщо всі елементи масиву менші за шукане число, програма має повертати значення `-1` (мінус один).

```
array[i] >= x
```

Алгоритм оформіть у вигляді функції

```
def bsearch_rightmost(array, x)
```

Гарантується, що вхідний масив array впорядкований за неспаданням.

Тестова програма розташована за посиланням:

<https://github.com/krenevych/algo/tree/master/labs/T3/Search/task3>

Завантажте всі файли, що містяться за цим посиланням, у одну папку. Реалізуйте згадану вище функцію, інтерфейс якої міститься у файлі `user.py`. Для перевірки правильності алгоритму запустіть файл `main.py`.

- 2.7. Знайдіть кількість входжень заданого числа x до масиву цілих чисел `array`. Алгоритм оформіть у вигляді функції

```
def counter(array, x)
```

Гарантується, що вхідний масив `array` впорядкований за неспаданням.

Тестова програма розташована за посиланням:

<https://github.com/krenevych/algo/tree/master/labs/T3/Search/task4>

Завантажте всі файли, що містяться за цим посиланням, у одну папку. Реалізуйте згадану вище функцію, інтерфейс якої міститься у файлі `user.py`. Для перевірки правильності алгоритму запустіть файл `main.py`.

- 2.8. Реалізуйте алгоритм лінійного пошуку всіх елементів списку, що є степенями двійки та визначте асимптотичну складність отриманого алгоритму.
- 2.9. Припустимо, що список містить елементи, для яких визначена операція \leq (менше або рівно) і елементи у цьому списку розташовані у порядку зростання. Скориставшись цією властивістю списку, запропонуйте алгоритм лінійного пошуку та оцініть його час виконання. Чи буде такий алгоритм оптимальнішим ніж класичний лінійний пошук?
- 2.10. Реалізуйте алгоритм бінарного пошуку з використанням рекурсії.

ЛАБОРАТОРНА РОБОТА 3. РОЗВ'ЯЗАННЯ РІВНЯНЬ МЕТОДОМ БІСКЦІЇ.

Контрольні запитання

- 3.1. Для чого використовується дійсний бінарний пошук? Наведіть приклад задачі. Чи є відмінність у реалізації дійсного бінарного пошуку від цілочисельного? Які основні підходи застосовуються для моменту завершення пошуку? У чому їхні переваги та недоліки у порівнянні один з одним? Наведіть алгоритм дійсного бінарного пошуку.

Завдання для аудиторної роботи

- 3.1. Для монотонної на відрізку $[a, b]$ функції f розв'яжіть рівняння

$$f(x) = c. \quad (3.1)$$

Реалізацію здійсніть у вигляді сукупності двох функцій описаних нижче. Функція

```
def solve(f, c, a, b):
```

Для неспадної на відрізку $[a, b]$ функції f розв'яже рівняння (3.1), а функція

```
def solve_decreasing(f, c, a, b)
```

відповідно для незростаючої на відрізку $[a, b]$ функції f розв'яже рівняння (3.1). Для визначення моменту завершення пошуку використайте всі три підходи (точність по аргументу, точність по значенню та безпосереднє сусідство двох чисел з плаваючою крапкою). Порівняйте отримані результати.

Тестова програма розташована за посиланням:

<https://github.com/kreneych/algo/tree/master/labs/T3/Search/task5>

Завантажте всі файли, що містяться за цим посиланням, у одну папку. Реалізуйте згадані вище функції, інтерфейси яких містяться у файлі `user.py`. Для перевірки правильності алгоритму запустіть файл `main.py`.

Завдання для самостійної роботи

- 3.2. <https://www.e-olymp.com/uk/problems/3968>

- 3.3. Знайдіть найменше $x \in [0, 10]$, що

$$f(x) = x^3 + x + 1 > 5$$

3.4. На відрізку $[1.6, 3]$ знайдіть корінь рівняння

$$\sin x = \frac{x}{3}.$$

3.5. На відрізку $[0, 2]$ знайдіть корінь рівняння

$$x^3 + 4x^2 + x - 6.$$

ЛАБОРАТОРНА РОБОТА 4. БІНАРНИЙ ПОШУК ПО ВІДПОВІДІ.

Контрольні запитання

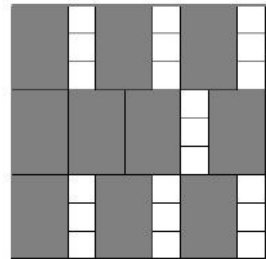
4.1.

Завдання для аудиторної роботи

4.1. Дипломи

<https://www.e-olymp.com/uk/problems/3969>

Коли Петя вчився в школі, він часто брав участь в олімпіадах з інформатики, математики та фізики. Так як він був досить здібним хлопчиком і старанно вчився, то на багатьох із цих олімпіад він отримував дипломи. До закінчення школи у нього накопичилося n дипломів, причому, як виявилось, всі вони мали однакові розміри: w - в ширину і h - у висоту.



Зараз Петя навчається в одному з кращих університетів і живе в гуртожитку зі своїми одногрупниками. Він вирішив прикрасити свою кімнату, повісивши на одну зі стін свої дипломи за шкільні олімпіади. Так як до бетонної стіни прикріпити дипломи досить важко, він вирішив купити спеціальну дошку з коркового дерева, щоб прикріпити її до стіни, а до неї - дипломи. Для того, щоб ця конструкція виглядала більш красиво, Петя хоче, щоб була квадратною і займала якомога менше місця на стіні. Кожен диплом повинен бути розміщений строго в прямокутнику w на h . дипломи забороняється повертати на 90 градусів. Прямокутники, які відповідають різним дипломам, не повинні мати спільних внутрішніх точок.

Потрібно написати програму, яка обчислить мінімальний розмір сторони дошки, яка буде потрібно Пете для розміщення всіх своїх дипломів.

Вхідні дані

Вхідний файл містить три цілих числа w , h , n ($1 \leq w, h, n \leq 109$).

Вихідні дані

У вихідний файл вивести відповідь до сформульованої задачі.

Вхідні дані #1

2 3 10

Вихідні дані #1

9

4.2. Дуже Легка Задача

<https://www.e-olymp.com/uk/problems/5102>

Сьогодні вранці журі вирішило додати у варіант олімпіади ще одну, Дуже Легку Задачу. Відповідальний секретар оргкомітету надрукував її умову в одному екземплярі, і тепер йому потрібно до початку олімпіади встигнути зробити ще n копій. У його розпорядженні є два ксерокси, один з яких копіює аркуш за x секунд, а другий за y . (Дозволяється використовувати як один ксерокс, так і обидва одночасно. Можна копіювати не лише з оригінала, але і з копії.) Допоможіть йому вияснити, який мінімальний час для цього потрібно.

Вхідні дані

Три натуральних числа n , x та y ($1 \leq n \leq 2 \cdot 10^8$, $1 \leq x, y \leq 10$).

Вихідні дані

Виведіть одне число – мінімальний час в секундах, необхідний для отримання n копій.

Вхідні дані #1

4 1 1

Вихідні дані #1

3

Вхідні дані #2

5 1 2

Вихідні дані #2

4

Завдання для самостійної роботи

4.3. Корови – в стійла

<https://www.e-olymp.com/uk/problems/5149>

4.4. Черепаха

<https://www.e-olymp.com/uk/problems/669>

4.5. Мутанти

<https://www.e-olymp.com/uk/problems/3970>

4.6. Рисосховище

<https://www.e-olymp.com/uk/problems/2254>

4.7. Мотузочки

<https://www.e-olymp.com/uk/problems/3967>

- 4.8. Роботи
<https://www.e-olymp.com/uk/problems/161>
- 4.9. Автобус
<https://www.e-olymp.com/uk/problems/6132>

ЛАБОРАТОРНА РОБОТА 5. Хешування та хеш-таблиці.

Контрольні запитання

5.1. Змодельуйте на дошці або на окремому листочку роботу з хеш-таблицею, що містить у ролі ключів

- а) цілі числа;
- б) рядкові літерали;
- с) як цілі числа та/або рядкові літерали.

Розгляньте обидва типи хеш-таблиць (з відкритою адресацією та ланцюжками).

5.2. Запропонуйте алгоритм видалення елементів з хеш-таблиці, що використовує метод лінійного зондування для розв'язання колізій.

5.3. Запропонуйте алгоритм збільшення слотів хеш-таблиці, що використовує лінійне зондування для розв'язання колізій, для випадку, якщо фактор завантаження таблиці перевищує деякий поріг.

Варіанти індивідуальних завдань

5.1.	https://www.e-olymp.com/uk/problems/4842
5.2.	https://www.e-olymp.com/uk/problems/1227
5.3.	https://www.e-olymp.com/uk/problems/7273
5.4.	https://www.e-olymp.com/uk/problems/131
5.5.	https://www.e-olymp.com/uk/problems/2035
5.6.	Розв'яжіть задачу https://informatics.msk.ru/mod/statements/view3.php?id=601&chapterid=745 використовуючи метод лінійного зондування для розв'язання колізій
5.7.	Розв'яжіть задачу https://informatics.msk.ru/mod/statements/view3.php?id=601&chapterid=745 використовуючи метод ланцюжків для розв'язання колізій
5.8.	https://www.e-olymp.com/ru/problems/3037
5.9.	https://www.e-olymp.com/uk/problems/1225

Допоміжна література по темі

- 5.1. А.П.Крєневич «Алгоритми і структури даних», §3.1
<https://1drv.ms/b/s!AkQ93-IlgoCHgpsqQvGRPDultn6aiw>
- 5.2. <http://aliev.me/runestone/SortSearch/Hashing.html>
- 5.3. https://csc.sibsutis.ru/sites/default/files/courses/pavu/S2_Lab3.pdf
- 5.4. <https://bitsofmind.wordpress.com/2008/07/28/introduction-in-hash-tables/>

ЛАБОРАТОРНА РОБОТА 6. Сортуння.

Контрольні запитання

6.1.

Завдання для аудиторної роботи

6.1.	https://www.e-olymp.com/uk/problems/2321
6.2.	https://www.e-olymp.com/uk/problems/2663
6.3.	https://www.e-olymp.com/uk/problems/5089
6.4.	https://www.e-olymp.com/uk/problems/2664

Варіанти індивідуальних завдань

6.1.	https://informatics.msk.ru/mod/statements/view3.php?id=1121&chapterid=767
6.2.	https://www.e-olymp.com/uk/problems/1303
6.3.	https://www.e-olymp.com/uk/problems/972
6.4.	https://www.e-olymp.com/uk/problems/4037
6.5.	https://www.e-olymp.com/uk/problems/1130
6.6.	https://www.e-olymp.com/uk/problems/1344
6.7.	https://www.e-olymp.com/uk/problems/1457
6.8.	https://www.e-olymp.com/uk/problems/1462
6.9.	https://www.e-olymp.com/uk/problems/2662
6.10.	https://www.e-olymp.com/ru/problems/4230
6.11.	https://www.e-olymp.com/uk/problems/1303
6.12.	https://www.e-olymp.com/ru/problems/3607

Допоміжна література по темі

- 6.1. А.П.Крєневич «Алгоритми і структури даних», §3.2
<https://1drv.ms/b/s!AkQ93-IIgoCHgpsqQvGRPDultn6aiw>
- 6.2. <http://aliev.me/runestone/SortSearch/TheBubbleSort.html>
- Булбашкове сортування:
<http://aliev.me/runestone/SortSearch/TheBubbleSort.html>
<https://www.youtube.com/watch?v=lyZQPjUT5B4>

Сортування вибором

<http://aliev.me/runestone/SortSearch/TheSelectionSort.html>

<https://www.youtube.com/watch?v=Ns4TPTC8whw&t=223s>

Сортування вставкою

<http://aliev.me/runestone/SortSearch/TheInsertionSort.html>

<https://www.youtube.com/watch?v=ROaIU379I3U&list=RDlyZQPjUT5B4&index=2>

Сортування злиттям

https://www.youtube.com/watch?v=XaqR3G_NVoo&index=3&list=RDlyZQPjUT5B

4

ЛАБОРАТОРНА РОБОТА 7. Рекурсія, повний перебір, Метод «розділай і володарюй».

Контрольні запитання

7.1.

Завдання для аудиторної роботи

7.1.	https://www.e-olymp.com/uk/problems/364
7.2.	https://www.e-olymp.com/uk/problems/1296
7.3.	https://www.e-olymp.com/uk/problems/1540
7.4.	https://www.e-olymp.com/uk/problems/1592
7.5.	https://www.e-olymp.com/uk/problems/1781

Варіанти індивідуальних завдань

7.1.	https://www.e-olymp.com/uk/problems/224
7.2.	https://www.e-olymp.com/uk/problems/317 – Множення Карацуби
7.3.	https://www.e-olymp.com/uk/problems/1266
7.4.	https://www.e-olymp.com/uk/problems/1524
7.5.	https://www.e-olymp.com/uk/problems/2000
7.6.	https://www.e-olymp.com/uk/problems/2633
7.7.	https://www.e-olymp.com/uk/problems/2764
7.8.	https://www.e-olymp.com/uk/problems/3606
7.9.	https://www.e-olymp.com/uk/problems/4746
7.10.	https://www.e-olymp.com/uk/problems/7031
7.11.	https://www.e-olymp.com/uk/problems/7227
7.12.	https://www.e-olymp.com/ru/problems/6

Допоміжна література по темі

- 7.1. А.П.Креневич «Алгоритми і структури даних», §4
<https://1drv.ms/b/s!AkQ93-llGoCHgpsqQvGRPDuln6aiw>
- 7.2. <https://logic.pdmi.ras.ru/~kulikov/simplealgscourse/lecture2-handout.pdf>
- 7.3. http://nord.org.ua/static/course/algo_2011/lecture3.pdf
- 7.4. <http://csaa.ru/algoritmy-razdeljaj-i-vlastvui/>
- 7.5. http://www.kalinin.ru/programming/alg/11_11_00.shtml

7.6.

<https://dic.academic.ru/dic.nsf/ruwiki/201612#.D0.A1.D0.B2.D1.8F.D0.B7.D1.8C.D1.81.D0.BA.D0.BE.D0.BD.D1.86.D0.B5.D0.BF.D1.86.D0.B8.D0.B5.D0.B9.C2.AB.D1.80.D0.B0.D0.B7.D0.B4.D0.B5.D0.BB.D1.8F.D0.B9.D0.B8.D0.B2.D0.BB.D0.B0.D1.81.D1.82.D0.B2.D1.83.D0.B9.C2.BB>

7.7. http://math.nsc.ru/LBRT/k4/or/or_part4.pdf

7.8. <https://www.matburo.ru/Examples/Files/Nazn3.pdf>

7.9.

<https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0%D0%BF%D0%B0%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%D1%80%D1%8E%D0%BA%D0%B7%D0%B0%D0%BA%D0%B0%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D0%B3%D1%96%D0%BB%D0%BE%D0%BA%D1%96%D0%BC%D0%B5%D0%B6>

7.10.

<https://ru.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5%D0%9A%D0%B0%D1%80%D0%B0%D1%86%D1%83%D0%B1%D1%8B> –Множення великих чисел

ЛАБОРАТОРНА РОБОТА 8. Стек та його застосування.

Контрольні запитання

8.1.

Завдання для аудиторної роботи

8.1.	https://www.e-olymp.com/uk/problems/1994
8.2.	https://www.e-olymp.com/uk/problems/4259
8.3.	https://www.e-olymp.com/uk/problems/2479
8.4.	https://www.e-olymp.com/uk/problems/7433
8.5.	https://www.e-olymp.com/uk/problems/855
8.6.	https://www.e-olymp.com/uk/problems/731
8.7.	https://www.e-olymp.com/uk/problems/5322
8.8.	

Варіанти індивідуальних завдань

8.1.	https://www.e-olymp.com/uk/problems/1376
8.2.	https://www.e-olymp.com/uk/problems/5205
8.3.	https://www.e-olymp.com/uk/problems/3013
8.4.	https://www.e-olymp.com/uk/problems/4438
8.5.	https://www.e-olymp.com/uk/problems/1801
8.6.	https://www.e-olymp.com/uk/problems/2039
8.7.	https://www.e-olymp.com/uk/problems/5539
8.8.	https://www.e-olymp.com/uk/problems/2702
8.9.	https://www.e-olymp.com/uk/problems/1008
8.10.	https://www.e-olymp.com/uk/problems/1377
8.11.	https://www.e-olymp.com/uk/problems/1394
8.12.	https://www.e-olymp.com/uk/problems/1073
8.13.	https://www.e-olymp.com/uk/problems/1515
8.14.	https://www.e-olymp.com/uk/problems/1549
8.15.	https://www.e-olymp.com/uk/problems/3837
	https://www.e-olymp.com/ru/problems/1776

Допоміжна література по темі

8.1.

ЛАБОРАТОРНА РОБОТА 9. Черги та зв'язні списки. Їхнє застосування

Контрольні запитання

9.1.

Завдання для аудиторної роботи

9.1.	https://www.e-olymp.com/uk/problems/971
9.2.	https://www.e-olymp.com/uk/problems/6125
9.3.	https://www.e-olymp.com/uk/problems/6126
9.4.	https://www.e-olymp.com/uk/problems/6127
9.5.	https://www.e-olymp.com/uk/problems/6128
9.6.	https://www.e-olymp.com/uk/problems/6129
9.7.	https://www.e-olymp.com/uk/problems/6130
9.8.	https://www.e-olymp.com/uk/problems/4847

Варіанти індивідуальних завдань

9.1.	https://www.e-olymp.com/uk/problems/3615
9.2.	https://www.e-olymp.com/uk/problems/1549
9.3.	https://www.e-olymp.com/uk/problems/1515
9.4.	https://www.e-olymp.com/uk/problems/2510
9.5.	https://www.e-olymp.com/uk/problems/3809
9.6.	https://www.e-olymp.com/uk/problems/4847
9.7.	https://www.e-olymp.com/uk/problems/694
9.8.	https://www.e-olymp.com/ru/problems/1228
9.9.	https://www.e-olymp.com/ru/problems/4005
9.10.	https://informatics.msk.ru/mod/statements/view.php?id=206
9.11.	https://informatics.msk.ru/mod/statements/view3.php?id=206&chapterid=112984
9.12.	https://informatics.msk.ru/mod/statements/view3.php?id=206&chapterid=53
9.13.	https://informatics.msk.ru/mod/statements/view3.php?id=601&chapterid=746

ЛАБОРАТОРНА РОБОТА 10. Дерева. Алгоритми на деревах.

Контрольні запитання

10.1.

Завдання для аудиторної роботи

10.1.	https://www.e-olymp.com/uk/problems/2317
10.2.	

Варіанти індивідуальних завдань

10.1.	https://www.e-olymp.com/uk/problems/2925
10.2.	https://www.e-olymp.com/uk/problems/4325
10.3.	https://www.e-olymp.com/uk/problems/4173
10.4.	https://www.e-olymp.com/uk/problems/5067
10.5.	https://www.e-olymp.com/uk/problems/2242
10.6.	https://www.e-olymp.com/uk/problems/2923
10.7.	https://www.e-olymp.com/uk/problems/3710
10.8.	https://www.e-olymp.com/uk/problems/3983
10.9.	https://www.e-olymp.com/uk/problems/3299

Допоміжна література по темі

10.1.

ЛАБОРАТОРНА РОБОТА 11. Бінарні дерева. Бінарні дерева пошуку.

Контрольні запитання

11.1.

Завдання для аудиторної роботи

11.1.	https://www.e-olymp.com/uk/problems/2312
11.2.	https://www.e-olymp.com/uk/problems/2316
11.3.	https://www.e-olymp.com/uk/problems/1513
11.4.	https://www.e-olymp.com/uk/problems/7466
11.5.	https://www.e-olymp.com/uk/problems/2314

Варіанти індивідуальних завдань

11.1.	Дерево синтаксичного розбору – http://aliev.me/runestone/Trees/ParseTree.html
11.2.	https://www.e-olymp.com/uk/problems/1846
11.3.	https://www.e-olymp.com/uk/problems/1516
11.4.	https://www.e-olymp.com/uk/problems/468
11.5.	https://www.e-olymp.com/uk/problems/2950
11.6.	https://www.e-olymp.com/uk/problems/3326

Допоміжна література по темі

11.1.

ЛАБОРАТОРНА РОБОТА 12. Бінарна купа та дерево відрізків.

Контрольні запитання

12.1.

Завдання для аудиторної роботи

12.1.	https://www.e-olymp.com/uk/problems/4481
12.2.	https://www.e-olymp.com/uk/problems/4482
12.3.	https://www.e-olymp.com/uk/problems/2919
12.4.	https://www.e-olymp.com/uk/problems/3737
12.5.	https://www.e-olymp.com/uk/problems/4039

Варіанти індивідуальних завдань

12.1.	https://www.e-olymp.com/uk/problems/5952
12.2.	https://www.e-olymp.com/uk/problems/695
12.3.	https://www.e-olymp.com/uk/problems/4496
12.4.	https://www.e-olymp.com/uk/problems/3318
12.5.	https://www.e-olymp.com/uk/problems/3329
12.6.	https://www.e-olymp.com/uk/problems/402
12.7.	https://www.e-olymp.com/uk/problems/4487
12.8.	https://www.e-olymp.com/uk/problems/4488
12.9.	https://www.e-olymp.com/uk/problems/4491
12.10.	https://www.e-olymp.com/uk/problems/4492
12.11.	https://www.e-olymp.com/uk/problems/4082
12.12.	https://www.e-olymp.com/uk/problems/3358
12.13.	https://www.e-olymp.com/uk/problems/5274

ЛАБОРАТОРНА РОБОТА 13. Алгоритми на не зважених графах.

Контрольні запитання

13.1.

Завдання для аудиторної роботи

13.1.	Зв'язність - https://www.e-olymp.com/uk/problems/982
13.2.	Обход в ширину - https://www.e-olymp.com/uk/problems/2401
13.3.	Перевірка на неорієнтовність – https://www.e-olymp.com/uk/problems/2470
13.4.	Від матриці суміжності до списку ребер – https://www.e-olymp.com/uk/problems/2471
13.5.	Витоки та стоки - https://www.e-olymp.com/uk/problems/3986
13.6.	Компоненти зв'язності - 2 – https://www.e-olymp.com/uk/problems/4816
13.7.	Найкоротша відстань - https://www.e-olymp.com/uk/problems/4852
13.8.	Найкоротший шлях - https://www.e-olymp.com/uk/problems/4853
13.9.	Підрахунок кількості ребер – https://www.e-olymp.com/uk/problems/5072
13.10.	Мультиребра - https://www.e-olymp.com/uk/problems/5073
13.11.	Степені вершин за списками ребер – https://www.e-olymp.com/uk/problems/5074
13.12.	Півстепені вершин за списками ребер – https://www.e-olymp.com/uk/problems/5075
13.13.	Кількість висячих вершин 1 – https://www.e-olymp.com/uk/problems/5080
13.14.	Степені вершин - https://www.e-olymp.com/uk/problems/5082

Варіанти індивідуальних завдань

13.1.	Маршрути в горах - https://www.e-olymp.com/uk/problems/122
13.2.	Стародавній рукопис - https://www.e-olymp.com/uk/problems/610
13.3.	Отримай дерево - https://www.e-olymp.com/uk/problems/978
13.4.	Топологічне сортування – https://www.e-olymp.com/uk/problems/1948
13.5.	Повний граф - https://www.e-olymp.com/uk/problems/3987

13.6.	Обхід у глибину - https://www.e-olymp.com/uk/problems/4000
13.7.	Геть списування! - https://www.e-olymp.com/uk/problems/4002
13.8.	Чи є цикл? - https://www.e-olymp.com/uk/problems/4004
13.9.	Числа - https://www.e-olymp.com/uk/problems/4007
13.10.	Підпал - https://www.e-olymp.com/uk/problems/4369
13.11.	Зв'язність графа - https://www.e-olymp.com/uk/problems/4374
13.12.	Магічна машинка - https://www.e-olymp.com/uk/problems/4850
13.13.	Регулярний граф - https://www.e-olymp.com/uk/problems/5076
13.14.	Напівповний граф - https://www.e-olymp.com/uk/problems/5077
13.15.	Транзитивність орієнтовного графа – https://www.e-olymp.com/uk/problems/5079
13.16.	Покриття шляхами - https://www.e-olymp.com/uk/problems/5299

Варіанти для командних завдань

13.1.	Рекурсія - https://www.e-olymp.com/uk/problems/553
13.2.	Ходи ферзем! - https://www.e-olymp.com/uk/problems/1098
13.3.	Сильная связность - https://www.e-olymp.com/uk/problems/2403
13.4.	Забавна гра - https://www.e-olymp.com/uk/problems/4050
13.5.	Хрещений батько - https://www.e-olymp.com/uk/problems/5366

Допоміжна література по темі

13.1.

ЛАБОРАТОРНА РОБОТА 14. Алгоритми на зважених графах

Контрольні запитання

14.1.

Завдання для аудиторної роботи

14.1.	Алгоритм Дейкстри - https://www.e-olymp.com/uk/problems/1365
14.2.	Найкоротший шлях - https://www.e-olymp.com/uk/problems/4856
14.3.	

Варіанти індивідуальних завдань

14.1.	Відстань між вершинами – https://www.e-olymp.com/uk/problems/625
14.2.	Флойд - 1 - https://www.e-olymp.com/uk/problems/974
14.3.	Флойд - https://www.e-olymp.com/uk/problems/975
14.4.	Флойд - існування - https://www.e-olymp.com/uk/problems/976
14.5.	Мінімальний каркас - https://www.e-olymp.com/uk/problems/981
14.6.	Червякові діри - https://www.e-olymp.com/uk/problems/1108
14.7.	Заправки - https://www.e-olymp.com/uk/problems/1388
14.8.	Автобуси - https://www.e-olymp.com/uk/problems/1389
14.9.	Форд-Беллман - https://www.e-olymp.com/uk/problems/1453
14.10.	Лабіринт знань - https://www.e-olymp.com/uk/problems/1454
14.11.	Цикл - https://www.e-olymp.com/uk/problems/1455
14.12.	Ліфти - https://www.e-olymp.com/uk/problems/2209
14.13.	Мінімальний каркас - https://www.e-olymp.com/uk/problems/3835
14.14.	Доставка кефірчика - https://www.e-olymp.com/uk/problems/4006
14.15.	

Варіанти для командних завдань

14.1.	Складний тест - https://www.e-olymp.com/uk/problems/626
14.2.	Мандри - https://www.e-olymp.com/uk/problems/2267
14.3.	Альтернативні шляхи - https://www.e-olymp.com/uk/problems/4637
14.4.	

ЛАБОРАТОРНА РОБОТА 15. Лабіринти.

Контрольні запитання

15.1.

Завдання для аудиторної роботи

15.1.	Площа кімнати - https://www.e-olymp.com/uk/problems/4001
15.2.	

Варіанти індивідуальних завдань

15.1.	Помста Лі Чака - https://www.e-olymp.com/uk/problems/88
15.2.	Підземелля - https://www.e-olymp.com/uk/problems/432
15.3.	Лінії - https://www.e-olymp.com/uk/problems/1060
15.4.	Фарбування лабіринту - https://www.e-olymp.com/uk/problems/1061
15.5.	Lines - https://www.e-olymp.com/uk/problems/1062
15.6.	Видалення клітинок - https://www.e-olymp.com/uk/problems/1063
15.7.	Шлях коня - https://www.e-olymp.com/uk/problems/1064
15.8.	Грядки - https://www.e-olymp.com/uk/problems/1065
15.9.	Переміщення коня - https://www.e-olymp.com/uk/problems/2820
15.10.	Іграшковий лабіринт - https://www.e-olymp.com/uk/problems/4452
15.11.	Вихід з лабіринту - https://www.e-olymp.com/uk/problems/4820
15.12.	Місце зустрічі змінити неможна – https://www.e-olymp.com/uk/problems/5069
15.13.	Лабіринт - https://www.e-olymp.com/uk/problems/5622
15.14.	Лабіринт - https://www.e-olymp.com/uk/problems/7215

Варіанти для командних завдань

15.1.	Виведи хомячків - https://www.e-olymp.com/uk/problems/213
15.2.	Сон - https://www.e-olymp.com/uk/problems/499
15.3.	Золота рибка – друге бажання Петрика – https://www.e-olymp.com/uk/problems/4634
15.4.	Гном і монети - https://www.e-olymp.com/uk/problems/7229

Допоміжна література по темі

15.1.

Список літератури та додаткових джерел

1. **Кренивч А.П.** Алгоритми та структури даних. Підручник. – К.: ВПЦ "Київський Університет", 2020. – 777 с.