

Дніпровський національний університет імені Олеся Гончара
Факультет прикладної математики
Кафедра обчислювальної математики та математичної кібернетики

ЗВІТ
про виконання лабораторної роботи 2
з дисципліни «Методи розв’язання задач дискретної оптимізації»

Виконав: студент гр. ПМ-22м-1
Щербак Роман

Дніпро
2023

1. Постановка задачі

Дано n предметів та рюкзак із місткістю W . $w = (w_1, w_2, \dots, w_n)$ — вектор ваг предметів, $v = (v_1, v_2, \dots, v_n)$ — вектор цінності предметів, $s = (s_1, s_2, \dots, s_n)$, де $s_i = j$, якщо i -ий предмет треба покласти до рюкзака не раніше j -ого предмета, $s_i = 0$, якщо для i -ого предмета не важливо, яким за порядком його буде покладено. Треба знайти такий вектор $x = (x_1, x_2, \dots, x_n)$, $x_i \in \{0, 1\}$, ($i = 1, 2, \dots, n$), при якому значення цільової функції буде максимальним:

$$\sum_{i=1}^n v_i x_i \rightarrow \max$$

при обмеженнях

$$\sum_{i=1}^n w_i x_i \leq W;$$

$$\sum_{i=1}^n w_i > W;$$

$$\forall i, j = \overline{1, n}: x_i = 1, s_i = j \Rightarrow s_j < i, \quad x_j \in \{0, 1\} \vee x_j = 0$$

2. Алгоритм розв'язку

Крок 1. Предмети групуються за послідовностями групування. Для кожної групи обчислюються загальна вага та загальна цінність за формулами:

$$W_i = \sum_{j=1}^p w_j, i = \overline{1, k};$$
$$V_i = \sum_{j=1}^p v_j, i = \overline{1, k},$$

де p — кількість предметів в групі (розмір групи). Якщо якийсь предмет не входить до жодної послідовності, то вважатимемо що цей предмет один в групі.

Крок 2. Якщо $k \neq 0$, то перевіряється виконання нерівності

$$W_i \leq W, i = \overline{1, k}$$

Кожні i -ту групу, для якої ця рівність не виконується, виключається з розгляду та $k = k - 1$.

Якщо $k = 0$, то розв'язку немає. Кінець алгоритму.

Крок 3. Розраховуємо питому цінність для груп предметів одним з двох способів:

1) $u_i = \frac{V_i}{W_i}, i = \overline{1, k}$

2) $u_i = \max_{j=\overline{1, p}} \{\frac{v_j}{w_j}\}, i = \overline{1, k}$

Впорядковуємо групи предметів за незростанням питомої цінності.

Крок 4. Будується бінарне дерево розв'язку. Для кожного вузла розраховується значення таких величин:

1) w — вага взятих речей;

2) v — цінність взятих речей;

3) u_b — оцінка зверху для значення цільової функції.

У випадку, якщо i -та група береться: $w = w + W_i; v = v + V_i$.

У випадку, якщо i -та група не береться: $w = w; v = v$.

$$u_b = v + (W - w) * u_{i+1}, 0 \leq i \leq k,$$

де u_{i+1} — питома цінність групи, наступної після i -ої в упорядкованому списку.

Для кореня дерева $w = 0; v = 0$. Якщо $i = k$, то $u_{i+1} = 0$.

Крок 5. Серед списку всіх висячих вершин дерева для наступного розгалуження вибирається та, якій відповідає розв'язок із найбільшою оцінкою зверху.

Після кожного розгалуження перевіряється виконання умови непереповненості рюкзака $w \leq W$. Якщо в деякій вершині ця умова не виконується, то ця вершина виключається з розв'язку. Якщо вершина з максимальною оцінкою зверху знаходиться на рівні $i < k$, перехід на крок 4.

Крок 6. Розв'язок знайдено. Значення цільової функції знаходиться у вершині з максимальним значенням u_b , тобто дорівнює значенню v цієї вершини, а вага — значенню w . Групи предметів, які потрібно взяти, визначаються за допомогою проходження від цієї вершини до кореня дерева.

3. Опис модулів програмного продукту

➤ Модуль main.py

Комбінує всі робочі модулі, здійснює обробку помилок у введенні, виводить результати роботи програми у файли (у вигляді побудованого бінарного дерева для критерію u_1 та u_2) та в консоль (у вигляді вектору розв'язків, ваги та вартости покладених у нього предметів) і час роботи алгоритму для обох заданих критеріїв.

➤ Модуль item_class.py

Містить клас Item, що реалізує весь необхідний функціонал предметів, доступних для вибору: створення предмета, надання йому індексу, ваги, ціни, з'єднання його з елементом, який має лежати до нього, призначення групи для цього предмету.

➤ Модуль graphic_interface.py

Утворює графічний інтерфейс, що дозволяє ввести параметри максимальної ваги рюкзака та вибрати метод отримання предметів (з файлу example.txt, example_1.txt або випадковим чином). На вхід отримує список параметрів із двох елементів (weight та choice), повертає словник, ключами якого є задані назви параметрів, а значеннями – введені користувачем у графічному інтерфейсі.

➤ Модуль graphic_interface_random.py

Розширює функціональність попереднього модуля для отримання параметрів випадкової генерації предметів: кількості предметів (можливі значення: від 5 до 100), проміжку можливих ваг

предметів (мінімальне можливе значення: 1, максимальне: 20), їх вартостей (мінімальне можливе значення: 1, максимальне: 100) та ймовірність, що предмет буде зв'язано з іншим предметів як його «наступника» (можливі значення: від 0 до 1). На вхід отримує список параметрів із шести елементів (item_amount, lower_weight, upper_weight, lower_value, upper_value, ordering_probability), повертає словник, ключами якого є задані назви параметрів, а значеннями – введені користувачем у графічному інтерфейсі.

➤ Модуль item_creation.py

Утворює випадковим чином або зчитує предмети з відповідного файлу, залежно від вибору, здійсненого користувачем. У випадку вибору випадковим чином модуль отримує параметри item_amount, lower_weight, upper_weight, lower_value, upper_value, ordering_probability, а у випадку вибору зчитування з файлу отримує параметр file_name. Повертає список утворених предметів.

➤ Модуль item_grouping.py

Групує утворені предмети за послідовностями, друкує їх у відповідних таблицях, що показують утворені предмети та утворені групи, групи, що не є заважкими для рюкзака, та порядок груп за критеріями u_1 та u_2 . На вхід отримує список предметів, повертає словник груп, де ключами є номери груп, а значеннями – списки предметів, що входять до кожної з тих груп. Цей словник разом із параметром максимальної ваги, яку можна покласти в рюкзак, подаються на вхід функції, що формує та повертає вказані вище таблиці, які потім передаються функції для друку цих таблиць у консоль.

➤ Модуль solving.py

Розв'язує задану задачу за допомогою алгоритму, що базується на методі гілок і меж. На вхід отримує список груп, упорядкований за відповідним критерієм, номер критерію, за яким треба розв'язувати задачу, максимальну вагу предметів, яку можна покласти в рюкзак. Повертає побудоване дерево, вектор результату, вагу та вартість предметів, покладених у рюкзак.

4. Інтерфейс користувача

Після запуску програми користувачу буде показано інтерфейс, де потрібно ввести максимальну вагу предметів, які можна покласти в рюкзак та вибрати з випадного списку метод отримання предметів: з файлу `example.txt`, `example_1.txt` або випадковим чином. Для запису параметрів необхідно натиснути кнопку «Записати параметри» та закрити вікно натисканням кнопки «Так» під запитанням «Завершуємо роботу з вікном?». У випадку правильного введення на консоль буде виведено словник, що міститиме введені параметри. Якщо введення було здійснено помилково, алгоритм не розпочне роботу та буде вказано, який із параметрів не було введено: якщо вагу не було введено чи було введено неправильно, то буде виведено «Вагу введено не числом.». Якщо вибір опції отримання параметрів не було здійснено, то буде виведено «Вибір опції отримання параметрів не було здійснено.».

Якщо помилок у введенні параметрів не було, алгоритм почне роботу (якщо задано вибір із одного з файлів) та видасть результат у форматі «критерій, для якого знайдено розв'язок, вектор результатів, вага предметів, вартість предметів, час роботи алгоритму), або вкаже, що рюкзак заповнити неможливо (якщо введено замалу вагу рюкзака) чи що «Вага рюкзака \geq вазі всіх предметів» (якщо введено зaveliku вагу рюкзака). У другому випадку буде вказано вектор розв'язку, що містить усі 1, а вага та вартість дорівнюють сумі ваг усіх груп.

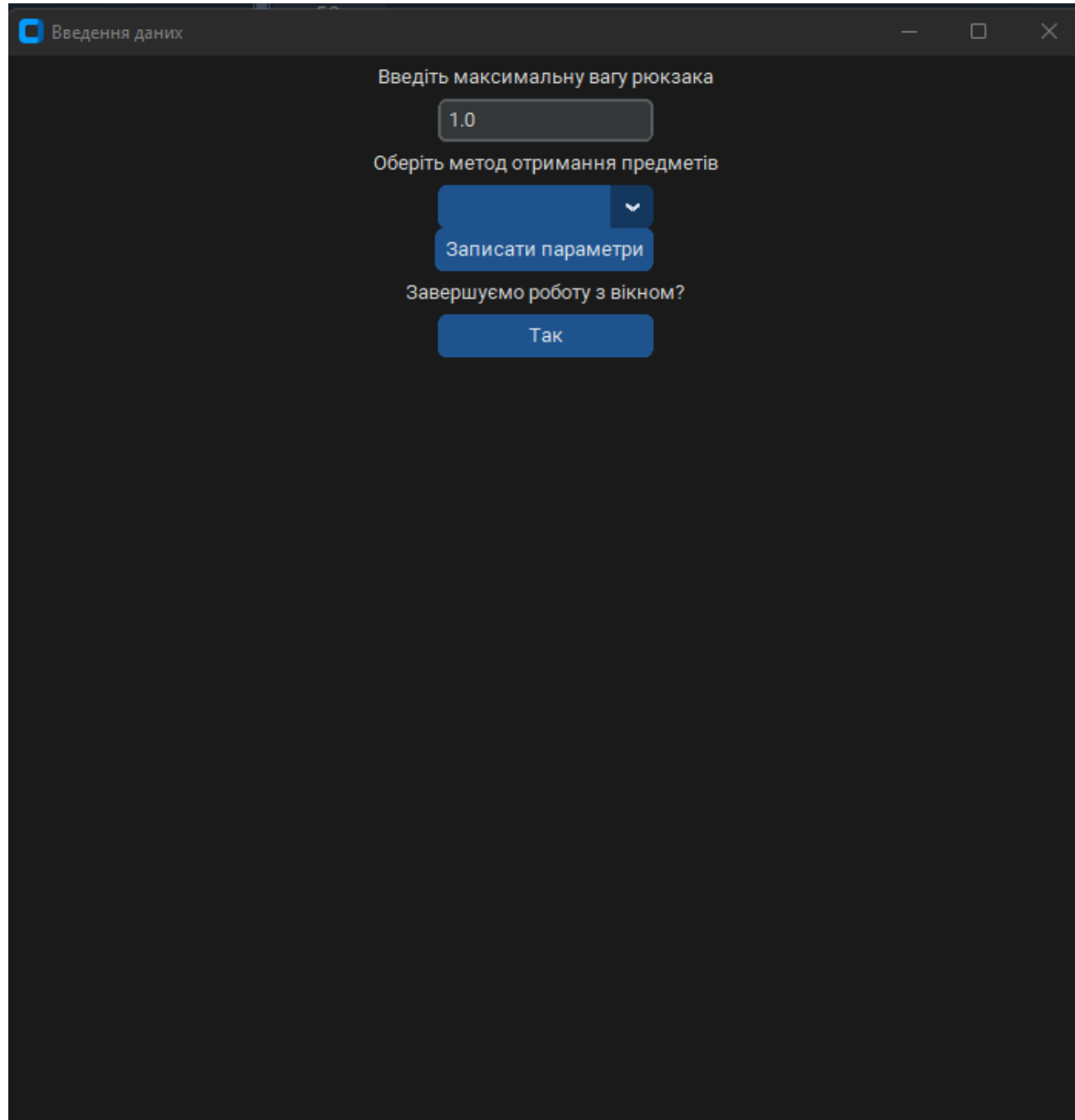
Якщо було вибрано отримання предметів випадковим чином, буде відкрито новий графічний інтерфейс, де потрібно за допомогою однобічного повзунка вибрати кількість предметів та ймовірність їх упорядкування, а за допомогою двобічних – можливі ваги та вартості їх. Запис параметрів та перехід до наступного кроку виконання програми здійснюється таким самим чином, як і для першого графічного

інтерфейсу. Якщо запис було здійснено, програма виведе в консоль параметри, які було введено за допомогою графічного інтерфейсу. Якщо не натиснути кнопку «Записати параметри», то буде видано повідомлення «Параметри не було записано». У випадку правильного введення параметрів алгоритм почне роботу та видасть результати таким самим чином, як було вказано в попередньому абзаці.

5. Код програми

https://github.com/Romchyk-S/MRZDO_Lab_2

6. Скриншоти роботи програми



Введення даних

Введіть максимальну вагу рюкзака

1.0

Оберіть метод отримання предметів

Записати параметри

Завершуємо роботу з вікном?

Так

Рисунок 1. Графічний інтерфейс для отримання початкових параметрів.

```

In [39]: runfile('E:/DNU/Marictpatupa/2 semester/MP3Д0/MRZD0_Lab_2/main.py', wdir='E:/DNU/Marictpatupa/2 semester/MP3Д0/MRZD0_Lab_2')
Reloaded modules: graphic_interface, graphic_interface_random, item_class, item_creation, item_groupping, solving

Exception in Tkinter callback
Traceback (most recent call last):
  File "E:\anaconda3\lib\tkinter\_init_.py", line 1921, in __call__
    return self.func(*args)
  File "E:\anaconda3\lib\site-packages\customtkinter\windows\widgets\ctk_entry.py", line 120, in _textvariable_callback
    if self._textvariable.get() == "":
  File "E:\anaconda3\lib\tkinter\_init_.py", line 593, in get
    return self._tk.getdouble(self._tk.globalgetvar(self._name))
_tkinter.TclError: expected floating-point number but got "b1.0"
Exception in Tkinter callback
Traceback (most recent call last):
  File "E:\anaconda3\lib\tkinter\_init_.py", line 1921, in __call__
    return self.func(*args)
  File "E:\anaconda3\lib\site-packages\customtkinter\windows\widgets\ctk_button.py", line 553, in _clicked
    self._command()
  File "E:\DNU\Marictpatupa\2 semester\MP3Д0\MRZD0_Lab_2\graphic_interface.py", line 86, in <lambda>
    submit_button = ctk.CTkButton(root, text = 'Записати параметри', command = lambda: button(working_parms))
  File "E:\DNU\Marictpatupa\2 semester\MP3Д0\MRZD0_Lab_2\graphic_interface.py", line 84, in button
    working_parms.parms_dict = {p:v.get() for p, v in zip(parms, variables)}
  File "E:\DNU\Marictpatupa\2 semester\MP3Д0\MRZD0_Lab_2\graphic_interface.py", line 84, in <dictcomp>
    working_parms.parms_dict = {p:v.get() for p, v in zip(parms, variables)}
  File "E:\anaconda3\lib\tkinter\_init_.py", line 593, in get
    return self._tk.getdouble(self._tk.globalgetvar(self._name))
_tkinter.TclError: expected floating-point number but got "b1.0"
Задано параметри
<property object at 0x000001F4DAF58E00>

ПОМИЛКИ У ВВЕДЕННІ:
Вагу введено не числом.

```

Рисунок 2. Випадок неправильного введення ваги рюкзака.

```

In [40]: runfile('E:/DNU/Marictpatupa/2 semester/MP3Д0/MRZD0_Lab_2/main.py', wdir='E:/DNU/Marictpatupa/2 semester/MP3Д0/MRZD0_Lab_2')
Reloaded modules: graphic_interface, graphic_interface_random, item_class, item_creation, item_groupping, solving

Задано параметри
<property object at 0x000001F4DB119710>

ПОМИЛКИ У ВВЕДЕННІ:
Вагу введено не числом.

```

Рисунок 3. Випадок ненажаття кнопки «Записати параметри».

```

In [41]: runfile('E:/DNU/Marictpatupa/2 semester/MP3Д0/MRZD0_Lab_2/main.py', wdir='E:/DNU/Marictpatupa/2 semester/MP3Д0/MRZD0_Lab_2')
Reloaded modules: graphic_interface, graphic_interface_random, item_class, item_creation, item_groupping, solving

Задано параметри
{'max_weight': 1.0, 'choice': ''}

Вибір опції отримання параметрів не було здійснено.

```

Рисунок 4. Випадок невибору параметрів зі списку.

```
In [46]: runfile('E:/DNU/Магістратура/2 semester/МРЗДО/1
Reloaded modules: graphic_interface, graphic_interface_

Задано параметри
{'max_weight': 7.0, 'choice': '3 файли example.txt'}
```

Предмети

№	Вага	Цінність	Попередник
1	7.0	56.0	None
2	6.0	35.0	None
3	5.0	30.0	2
4	4.0	25.0	None
5	3.0	20.0	None
6	2.0	15.0	5
7	1.0	10.0	6

Таблиця груп

№	Предмети	Вага	Цінність
1	[1]	7.0	56.0
2	[2, 3]	11.0	65.0
3	[4]	4.0	25.0
4	[5, 6, 7]	6.0	45.0

Таблиця груп без надто важких

№	Предмети	Вага	Цінність
1	[1]	7.0	56.0
3	[4]	4.0	25.0
4	[5, 6, 7]	6.0	45.0

Таблиця груп із відотною цінністю

№	Предмети	Вага	Цінність	u_1	u_2
1	[1]	7.0	56.0	8.0	8.0
3	[4]	4.0	25.0	6.25	6.25
4	[5, 6, 7]	6.0	45.0	7.5	10.0

Рисунок 5. Випадок правильного вибору всіх параметрів для зчитування з файлу example.txt (1).

```

Таблиця груп, сортована за u_1
+---+---+---+---+---+---+
| № | Предмети | Вага | Цінність | u_1 | u_2 |
+---+---+---+---+---+---+
| 1 | [1]       | 7.0  | 56.0     | 8.0  | 8.0  |
| 4 | [5, 6, 7] | 6.0  | 45.0     | 7.5  | 10.0 |
| 3 | [4]       | 4.0  | 25.0     | 6.25 | 6.25 |
+---+---+---+---+---+---+

Таблиця груп, сортована за u_2
+---+---+---+---+---+---+
| № | Предмети | Вага | Цінність | u_1 | u_2 |
+---+---+---+---+---+---+
| 4 | [5, 6, 7] | 6.0  | 45.0     | 7.5  | 10.0 |
| 1 | [1]       | 7.0  | 56.0     | 8.0  | 8.0  |
| 3 | [4]       | 4.0  | 25.0     | 6.25 | 6.25 |
+---+---+---+---+---+---+

Сформовано груп:
3

Розв'язок для критерію u_1
Результат: [1, 0, 0, 0, 0, 0, 0]
W = 7.0
V = 56.0
Розв'язано за 0.0 секунд

Розв'язок для критерію u_2
Результат: [1, 0, 0, 0, 0, 0, 0]
W = 7.0
V = 56.0
Розв'язано за 0.0 секунд

```

Рисунок 6. Випадок правильного вибору всіх параметрів для зчитування з файлу example.txt (2).

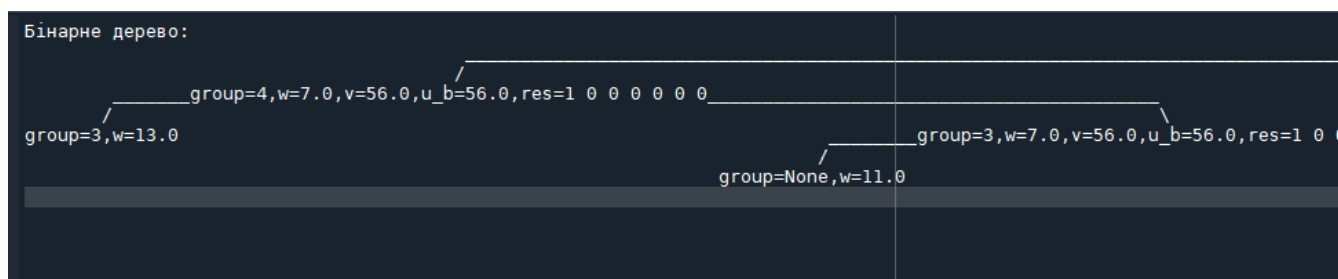


Рисунок 7. Частина бінарного дерева для прикладу з файлу example.txt за критерієм u_1 .

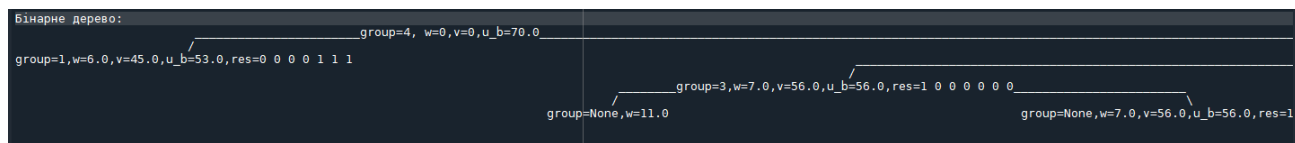


Рисунок 8. Частина бінарного дерева для прикладу з файлу example.txt за критерієм u_2 .

```
In [42]: runfile('E:/DNU/Marictpatupa/2 semester/MP3ДО/MRZD
Reloaded modules: graphic_interface, graphic_interface_ran

Задано параметри
{'max_weight': 12.0, 'choice': '3 файлу example_1.txt'}

Предмети
+---+
| № | Вага | Цінність | Попередник |
+---+
| 1 | 6.29 | 55.37 | None |
| 2 | 5.19 | 55.63 | None |
| 3 | 6.73 | 53.64 | 7 |
| 4 | 4.35 | 54.91 | None |
| 5 | 4.26 | 58.91 | None |
| 6 | 6.49 | 56.43 | None |
| 7 | 6.32 | 58.49 | None |
| 8 | 4.73 | 56.29 | 1 |
+---+

Таблиця груп
+---+
| № | Предмети | Вага | Цінність |
+---+
| 1 | [1, 8] | 11.02 | 111.66 |
| 2 | [2] | 5.19 | 55.63 |
| 3 | [7, 3] | 13.05 | 112.13 |
| 4 | [4] | 4.35 | 54.91 |
| 5 | [5] | 4.26 | 58.91 |
| 6 | [6] | 6.49 | 56.43 |
+---+

Таблиця груп без надто важких
+---+
| № | Предмети | Вага | Цінність |
+---+
| 1 | [1, 8] | 11.02 | 111.66 |
| 2 | [2] | 5.19 | 55.63 |
| 4 | [4] | 4.35 | 54.91 |
| 5 | [5] | 4.26 | 58.91 |
```

Рисунок 9. Випадок правильного вибору всіх параметрів для зчитування з файлу example_1.txt (1).

Таблиця груп без надто важких					
№	Предмети	Вага	Цінність		
1	[1, 8]	11.02	111.66		
2	[2]	5.19	55.63		
4	[4]	4.35	54.91		
5	[5]	4.26	58.91		
6	[6]	6.49	56.43		

Таблиця груп із відотною цінністю					
№	Предмети	Вага	Цінність	u_1	u_2
1	[1, 8]	11.02	111.66	10.13	11.9
2	[2]	5.19	55.63	10.72	10.72
4	[4]	4.35	54.91	12.62	12.62
5	[5]	4.26	58.91	13.83	13.83
6	[6]	6.49	56.43	8.69	8.69

Таблиця груп, сортована за u_1					
№	Предмети	Вага	Цінність	u_1	u_2
5	[5]	4.26	58.91	13.83	13.83
4	[4]	4.35	54.91	12.62	12.62
2	[2]	5.19	55.63	10.72	10.72
1	[1, 8]	11.02	111.66	10.13	11.9
6	[6]	6.49	56.43	8.69	8.69

Таблиця груп, сортована за u_2					
№	Предмети	Вага	Цінність	u_1	u_2
5	[5]	4.26	58.91	13.83	13.83
4	[4]	4.35	54.91	12.62	12.62
1	[1, 8]	11.02	111.66	10.13	11.9
2	[2]	5.19	55.63	10.72	10.72
6	[6]	6.49	56.43	8.69	8.69

Сформовано груп:
5

Рисунок 10. Випадок правильного вибору всіх параметрів для зчитування з файлу example_1.txt (2).


```
Розв'язок для критерію  $u_1$   
Результат: [0, 0, 0, 0, 1, 1, 0, 0]  
W = 10.75  
V = 115.34  
Розв'язано за 0.01 секунд
```



```
Розв'язок для критерію  $u_2$   
Результат: [0, 0, 0, 0, 1, 1, 0, 0]  
W = 10.75  
V = 115.34  
Розв'язано за 0.01 секунд
```

Рисунок 11. Випадок правильного вибору всіх параметрів для зчитування з файлу example_1.txt (3).

Рисунок 12. Частина бінарного дерева для прикладу з файлу example_1.txt за критерієм u_1 .

Рисунок 13. Частина бінарного дерева для прикладу з файлу example_1.txt за критерієм u_2 .



Рисунок 14. Графічний інтерфейс для отримання параметрів випадкової генерації предметів.

```

In [47]: runfile('E:/DNU/Марієтратура/2 semester/MP3/
Reloaded modules: graphic_interface, graphic_interfa

Задано параметри
{'max_weight': 12.0, 'choice': 'Випадковим чином'}

Параметри випадкової генерації предметів:
<property object at 0x000001F4DAFCD210>

Параметри не було записано.

```

Рисунок 15. Випадок незапису параметрів випадкової генерації предметів.

```

Задано параметри
{'max_weight': 12.0, 'choice': 'Випадковим чином'}

Параметри випадкової генерації предметів:
{'item_amount': 16, 'lower_weight': 2.0058823529411764, 'upper_weight': 3.347058823529412, 'lower_value': 13.520588235294117, 'upper_value': 19.63529411764706,
'ordering_probability': 0.27}

Предмети
+-----+
| № | Вага | Цінність | Попередник |
+-----+
| 1 | 3.18 | 18.51 | None |
| 2 | 3.2 | 17.16 | 10 |
| 3 | 3.22 | 17.36 | None |
| 4 | 3.04 | 15.55 | None |
| 5 | 2.82 | 15.26 | 1 |
| 6 | 2.84 | 18.73 | None |
| 7 | 2.16 | 18.07 | None |
| 8 | 2.91 | 14.67 | None |
| 9 | 2.22 | 18.08 | None |
| 10 | 2.02 | 18.81 | None |
| 11 | 2.48 | 18.57 | None |
| 12 | 2.47 | 16.27 | None |
| 13 | 3.26 | 17.95 | None |
| 14 | 3.08 | 19.19 | 4 |
| 15 | 2.76 | 15.37 | None |
| 16 | 2.15 | 18.56 | None |
+-----+

Таблиця груп
+-----+
| № | Предмети | Вага | Цінність |
+-----+
| 1 | [1, 5] | 6.0 | 33.77 |
| 2 | [10, 2] | 5.22 | 35.97 |
| 3 | [3] | 3.22 | 17.36 |
| 4 | [4, 14] | 6.12 | 34.74 |
| 5 | [6] | 2.84 | 18.73 |
| 6 | [7] | 2.16 | 18.07 |
| 7 | [8] | 2.91 | 14.67 |
| 8 | [9] | 2.22 | 18.08 |
| 9 | [11] | 2.48 | 18.57 |
| 10 | [12] | 2.47 | 16.27 |
| 11 | [13] | 3.26 | 17.95 |
| 12 | [15] | 2.76 | 15.37 |
| 13 | [16] | 2.15 | 18.56 |
+-----+

```

Рисунок 16. Випадок правильного вибору всіх параметрів для випадкової генерації предметів (1).

Таблиця груп без надто важких				
№	Предмети	Вага	Цінність	
1	[1, 5]	6.0	33.77	
2	[10, 2]	5.22	35.97	
3	[3]	3.22	17.36	
4	[4, 14]	6.12	34.74	
5	[6]	2.84	18.73	
6	[7]	2.16	18.07	
7	[8]	2.91	14.67	
8	[9]	2.22	18.08	
9	[11]	2.48	18.57	
10	[12]	2.47	16.27	
11	[13]	3.26	17.95	
12	[15]	2.76	15.37	
13	[16]	2.15	18.56	

Таблиця груп із відносною цінністю					
№	Предмети	Вага	Цінність	u_1	u_2
1	[1, 5]	6.0	33.77	5.63	5.82
2	[10, 2]	5.22	35.97	6.89	9.31
3	[3]	3.22	17.36	5.39	5.39
4	[4, 14]	6.12	34.74	5.68	6.23
5	[6]	2.84	18.73	6.6	6.6
6	[7]	2.16	18.07	8.37	8.37
7	[8]	2.91	14.67	5.04	5.04
8	[9]	2.22	18.08	8.14	8.14
9	[11]	2.48	18.57	7.49	7.49
10	[12]	2.47	16.27	6.59	6.59
11	[13]	3.26	17.95	5.51	5.51
12	[15]	2.76	15.37	5.57	5.57
13	[16]	2.15	18.56	8.63	8.63

Таблиця груп, сортована за u_1					
№	Предмети	Вага	Цінність	u_1	u_2
13	[16]	2.15	18.56	8.63	8.63
6	[7]	2.16	18.07	8.37	8.37
8	[9]	2.22	18.08	8.14	8.14
9	[11]	2.48	18.57	7.49	7.49

Рисунок 17. Випадок правильного вибору всіх параметрів для випадкової генерації предметів (2).

6	[7]	2.16	18.07	8.37	8.37
8	[9]	2.22	18.08	8.14	8.14
9	[11]	2.48	18.57	7.49	7.49
2	[10, 2]	5.22	35.97	6.89	9.31
5	[6]	2.84	18.73	6.6	6.6
10	[12]	2.47	16.27	6.59	6.59
4	[4, 14]	6.12	34.74	5.68	6.23
1	[1, 5]	6.0	33.77	5.63	5.82
12	[15]	2.76	15.37	5.57	5.57
11	[13]	3.26	17.95	5.51	5.51
3	[3]	3.22	17.36	5.39	5.39
7	[8]	2.91	14.67	5.04	5.04

Таблиця груп, сортована за u_2

№	Предмети	Вага	Цінність	u_1	u_2
2	[10, 2]	5.22	35.97	6.89	9.31
13	[16]	2.15	18.56	8.63	8.63
6	[7]	2.16	18.07	8.37	8.37
8	[9]	2.22	18.08	8.14	8.14
9	[11]	2.48	18.57	7.49	7.49
5	[6]	2.84	18.73	6.6	6.6
10	[12]	2.47	16.27	6.59	6.59
4	[4, 14]	6.12	34.74	5.68	6.23
1	[1, 5]	6.0	33.77	5.63	5.82
12	[15]	2.76	15.37	5.57	5.57
11	[13]	3.26	17.95	5.51	5.51
3	[3]	3.22	17.36	5.39	5.39
7	[8]	2.91	14.67	5.04	5.04

Сформовано груп:
13

Розв'язок для критерію u_1
Результат: [0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1]
W = 11.85
V = 92.01
Розв'язано за 0.02 секунд

Розв'язок для критерію u_2
Результат: [0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1]
W = 11.85
V = 92.01
Розв'язано за 0.03 секунд

Рисунок 18. Випадок правильного вибору всіх параметрів для випадкової генерації предметів (3).

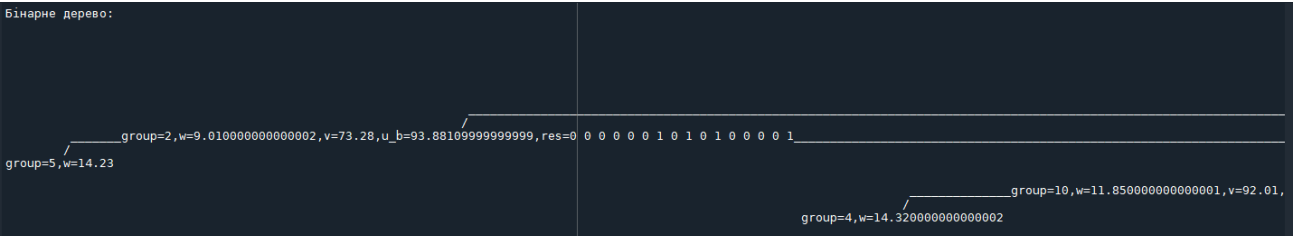


Рисунок 19. Частина бінарного дерева для прикладу випадкової генерації предметів за критерієм u_1 .

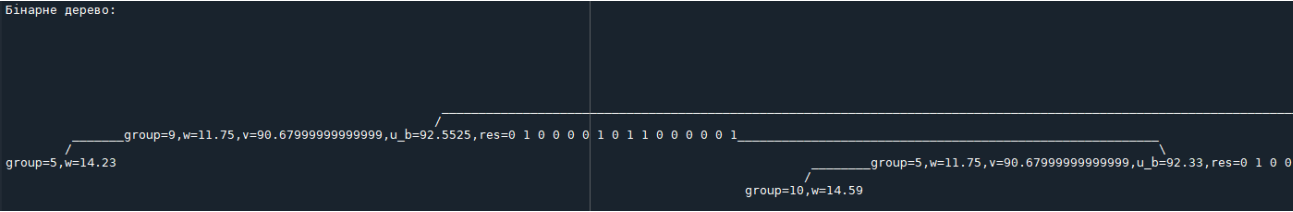


Рисунок 20. Частина бінарного дерева для прикладу випадкової генерації предметів за критерієм u_1 .

Задано параметри
{'max_weight': 1.0, 'choice': '3 файлу example_1.txt'}

Предмети

№	Вага	Цінність	Попередник
1	6.29	55.37	None
2	5.19	55.63	None
3	6.73	53.64	7
4	4.35	54.91	None
5	4.26	58.91	None
6	6.49	56.43	None
7	6.32	58.49	None
8	4.73	56.29	1

Таблиця груп

№	Предмети	Вага	Цінність
1	[1, 8]	11.02	111.66
2	[2]	5.19	55.63
3	[7, 3]	13.05	112.13
4	[4]	4.35	54.91
5	[5]	4.26	58.91
6	[6]	6.49	56.43

Таблиця груп без надто важких

№	Предмети	Вага	Цінність

Сформовано груп:

0

Рюкзак заповнити неможливо.

Рисунок 21. Випадок замалої місткості рюкзака (для прикладу з файлу example_1.txt).

```
Задано параметри
{'max_weight': 1000.0, 'choice': 'З файлу example_1.txt'}
```

Предмети			
№	Вага	Цінність	Попередник
1	6.29	55.37	None
2	5.19	55.63	None
3	6.73	53.64	7
4	4.35	54.91	None
5	4.26	58.91	None
6	6.49	56.43	None
7	6.32	58.49	None
8	4.73	56.29	1

Таблиця груп			
№	Предмети	Вага	Цінність
1	[1, 8]	11.02	111.66
2	[2]	5.19	55.63
3	[7, 3]	13.05	112.13
4	[4]	4.35	54.91
5	[5]	4.26	58.91
6	[6]	6.49	56.43

Таблиця груп без надто важких			
№	Предмети	Вага	Цінність
1	[1, 8]	11.02	111.66
2	[2]	5.19	55.63
3	[7, 3]	13.05	112.13
4	[4]	4.35	54.91
5	[5]	4.26	58.91
6	[6]	6.49	56.43

Таблиця груп із відносною цінністю					
№	Предмети	Вага	Цінність	u_1	u_2
1	[1, 8]	11.02	111.66	10.13	11.9
2	[2]	5.19	55.63	10.72	10.72

Рисунок 22. Випадок завеликої місткості рюкзака (для прикладу з файлу example_1.txt) (1).

Таблиця груп із відотною цінністю						
№	Предмети	Вага	Цінність	u_1	u_2	
1	[1, 8]	11.02	111.66	10.13	11.9	
2	[2]	5.19	55.63	10.72	10.72	
3	[7, 3]	13.05	112.13	8.59	9.25	
4	[4]	4.35	54.91	12.62	12.62	
5	[5]	4.26	58.91	13.83	13.83	
6	[6]	6.49	56.43	8.69	8.69	

Таблиця груп, сортована за u_1						
№	Предмети	Вага	Цінність	u_1	u_2	
5	[5]	4.26	58.91	13.83	13.83	
4	[4]	4.35	54.91	12.62	12.62	
2	[2]	5.19	55.63	10.72	10.72	
1	[1, 8]	11.02	111.66	10.13	11.9	
6	[6]	6.49	56.43	8.69	8.69	
3	[7, 3]	13.05	112.13	8.59	9.25	

Таблиця груп, сортована за u_2						
№	Предмети	Вага	Цінність	u_1	u_2	
5	[5]	4.26	58.91	13.83	13.83	
4	[4]	4.35	54.91	12.62	12.62	
1	[1, 8]	11.02	111.66	10.13	11.9	
2	[2]	5.19	55.63	10.72	10.72	
3	[7, 3]	13.05	112.13	8.59	9.25	
6	[6]	6.49	56.43	8.69	8.69	

Сформовано груп:
6

Вага рюкзака \geq вазі всіх предметів.
Результат: [1, 1, 1, 1, 1, 1, 1, 1]
W = 44.36
V = 449.67

Рисунок 23. Випадок завеликої місткості рюкзака (для прикладу з файлу example_1.txt) (2).

7. Висновок

У ході роботи було розглянуто задачу про рюкзак із частковим упорядкуванням та алгоритм її розв'язання, що базується на методі гілок та меж, написано програму, що реалізує цей алгоритм мовою Python, протестовано код на прикладах.