

Лабораторная работа 4

Черепенников Роман, 3 курс 8 группа

Задача о рюкзаке (англ. Knapsack problem) — дано n предметов, предмет i имеет массу $w_i > 0$ и стоимость $p_i > 0$. Необходимо выбрать из этих предметов такой набор, чтобы суммарная масса не превосходила заданной величины W (вместимость рюкзака), а суммарная стоимость была максимальна.

Рассмотрим задачу **Неограниченный рюкзак** (англ. Unbounded Knapsack Problem), в которой любой предмет может быть выбран любое количество раз.

Формулировка Задачи

Каждый предмет может быть выбран любое число раз. Задача выбрать количество x_i предметов каждого типа так, чтобы

максимизировать общую стоимость: $\sum_{i=1}^n p_i x_i$;

выполнялось условие совместности: $\sum_{i=1}^n w_i x_i \leq W$;

где $x_i \geq 0$ целое, для всех $i = 1, 2, \dots, n$.

Вариант 21

$$\begin{aligned} & \max 5x_1 + 8x_2 + 15x_3 \\ & \text{s.t. } 3x_1 + 4x_2 + 5x_3 \leq 10 \\ & x_1, x_2, x_3 \geq 0 - \text{целые.} \end{aligned}$$

Листинг программы:

```
import numpy as np
from typing import Tuple

def knapsack(weights: np.ndarray, v: np.ndarray, W: int) -> Tuple[int, np.ndarray]:
    if weights.shape != v.shape:
        raise ValueError('w and v should have same shape')

    n = weights.shape[0]
    f = np.zeros((n + 1, W + 1))
    p = np.zeros((n + 1, W + 1))

    for i in range(1, n + 1):
        for w in range(1, W + 1):
            if w >= weights[i - 1]:
                f[i, w] = max([f[i - 1, w], f[i, w - weights[i - 1]] + v[i - 1]])
            else:
                f[i, w] = f[i - 1, w]

        p[i, w] = int(f[i, w] != f[i - 1, w])

    # restoring objects
    taken = np.zeros_like(weights)
    w = W
    i = n
    while i != 0 and w != 0:
        if p[i, w] == 1:
            taken[i - 1] += 1
            w -= weights[i - 1]
        else:
            i -= 1

    return f[n, W], taken
```

```
if __name__ == "__main__":  
    w = np.array([3, 4, 5])  
    W = 10  
    v = np.array([5, 8, 15])  
    target, taken = knapsack(w, v, W)  
    print(f'Оптимальное решение задачи: {target}')
```

```
    print(f'x1 = {taken[0]}, x2 = {taken[1]}, x3 = {taken[2]}')
```

Вывод программы:

Оптимальное решение задачи: 30.0

x1 = 0, x2 = 0, x3 = 2