

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра информационных систем управления

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 4
ВАРИАНТ 5

Выполнил:

Карпович Артём Дмитриевич
студент 3 курса 7 группы

Преподаватель:

Кваша Дарья Юрьевна

Минск, 2024

Задача о рюкзаке.

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:

$$\sum_{i=0}^n p_i x_i$$

налось условие совместности:

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

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

Задача

Решите следующую задачу о рюкзаке, записав результаты рекурсии динамического программирования в таблицу. Напишите алгоритм обратного хода.

$$\max 3x_1 + 8x_2 + 14x_3$$

$$\text{s.t. } 2x_1 + 3x_2 + 5x_3 \leq 9$$

$$x_1, x_2, x_3 \geq 0.$$

Программная реализация.

```
[65]: import pandas as pd
import numpy as np

def unbounded_knapsack(n, w, v, W):
    f = [[0] * (W + 1) for _ in range(n + 1)]
    p = [[0] * (W + 1) for _ in range(n + 1)]

    for weight in range(1, W + 1):
        for i in range(1, n + 1):
            f[i][weight] = f[i - 1][weight]
            p[i][weight] = 0
            if w[i - 1] <= weight:
                if v[i - 1] + f[i][weight - w[i - 1]] > f[i][weight]:
                    f[i][weight] = v[i - 1] + f[i][weight - w[i - 1]]
                    p[i][weight] = 1

    print("Таблица f:")
    for i in range(W + 1):
```

```

        for weight in range(1, n + 1):
            print(f"{f[weight][i]:2}", end=" ")
        print()

    print("Таблица p:")
    for i in range(W + 1):
        for weight in range(1, n + 1):
            print(f"{p[weight][i]:2}", end=" ")
        print()

    return f[n][W]

w = [3, 8, 14]
v = [2, 3, 5]
W = 9

max_value = unbounded_knapsack(len(w), w, v, W)

print("Максимальная стоимость:", max_value)

```

Таблица f:

0	0	0
0	0	0
0	0	0
2	2	2
2	2	2
2	2	2
4	4	4
4	4	4
4	4	4
6	6	6

Таблица p:

0	0	0
0	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0

Максимальная стоимость: 6

Оптимальное решение задачи: $f_3(9) = 6$.

Обратный ход

Стартуем с $x_1 = x_2 = x_3 = 0$.

$p_3(9) = 0 \Rightarrow x_3 = x_3 + 0$. Объем не изменился, осталось 9 единиц.

$p_2(9) = 0 \Rightarrow x_2 = x_2 + 0$; Объем не изменился, осталось 9 единиц.

$p_1(9) = 1 \Rightarrow x_1 = x_1 + 1$; Объем изменился, осталось $9-3=6$ единиц.

$p_1(6) = 1 \Rightarrow x_1 = x_1 + 1$; Объем изменился, осталось $6-3=3$ единиц.

$p_1(3) = 1 \Rightarrow x_1 = x_1 + 1$; Объем изменился, осталось $3-3=0$ единицы.

Стоп. Оптимальное решение задачи: $x_1 = 3, x_2 = 0, x_3 = 0$.