Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Звіт**

з лабораторної роботи № 9 з дисципліни

Алгоритми та структури даних 2. Структури даних»

«Жадібні алгоритми»

Варіант **3**

**Виконав студент:** ІП-з21 Гавриленко Даяна Юріївна

**Перевірив:** Халус Олена Андріївна

Київ 2023

**9.1 Мета роботи**

Мета роботи – на практиці навчитися використовувати динамічне програмування.

**9.2 Постановка задачі**

Дано n різних предметів, про які відомі їх розмір, або вага, wi та вартість vi. Є рюкзак, в який необхідно покласти ці предмети. Для рюкзака відома його місткість (сумарний розмір, або вага, предметів, що можуть бути розміщені у рюкзаку) — W. Необхідно відібрати таку множину предметів S серед усіх заданих предметів, що (1) їх сумарна розмірність не перевищує місткість рюкзака W та (2) сумарна вартість предметів в множині S є максимально можливою серед усіх інших множин.  
Для розв'язку задачі про рюкзак методом динамічного програмування необхідно спочатку сформулювати розв'язок початкової задачі через розв'язання задач меншої розмірності. Для цього позначимо через S — максимальну вартість предметів, які можна розмістити у рюкзаку. Припустимо, що останній предмет No n належить S, тоді S – {n} — оптимальний розв'язок для перших n – 1 предметів і місткості рюкзака (W – wn). Тепер можна сформулювати рекурсивне правило для отримання розв'язку задачі. Позначимо через V(i, x) найбільшу вартість, таку що (1) розглянуті тільки перші i предметів та (2) загальний розмір предметів не перевищує x. Тоді:

V(i, x) = max{ V(i – 1, x), V(i – 1, x – wi) + vi } (\*) Примітка: у випадку коли wi > x, V(i, x) = V(i – 1, x) .  
Тож, алгоритм динамічного програмування повинен перебрати підзадачі всіх можливих розмірностей, які визначаються двома змінними:  
• i – кількість предметів (1, ..., n)  
• x – місткість рюкзака (1, ..., W)

**9.3 Псевдокод алгоритму**

функція Рюкзак(W, weights, values, n):

створити двовимірний масив dp розміром (n + 1) x (W + 1)

для i від 0 до n:

для w від 0 до W:

якщо i = 0 або w = 0:

dp[i, w] = 0

інакше, якщо weights[i - 1] <= w:

dp[i, w] = max(values[i - 1] + dp[i - 1, w - weights[i - 1]], dp[i - 1, w])

інакше:

dp[i, w] = dp[i - 1, w]

повернути dp[n, W]

зчитати W, n з вхідного файлу

створити порожній масив values розміром n

створити порожній масив weights розміром n

для i від 0 до n - 1:

зчитати vi, wi з вхідного файлу

values[i] = vi

weights[i] = wi

результат = Рюкзак(W, weights, values, n)

записати результат у вихідний файл

**9.4** **Програмна реалізація алгоритму**

**9.4.1 Вихідний код на мові C#**

using System;  
using System.IO;  
  
class KnapsackProblem  
{  
 static int Knapsack(int W, int[] weights, int[] values, int n)  
 {  
 int[,] dp = new int[n + 1, W + 1];  
  
 for (int i = 0; i <= n; i++)  
 {  
 for (int w = 0; w <= W; w++)  
 {  
 if (i == 0 || w == 0)  
 dp[i, w] = 0;  
 else if (weights[i - 1] <= w)  
 dp[i, w] = Math.Max(values[i - 1] + dp[i - 1, w - weights[i - 1]], dp[i - 1, w]);  
 else  
 dp[i, w] = dp[i - 1, w];  
 }  
 }  
  
 return dp[n, W];  
 }  
  
 static void Main()  
 {  
 string[] lines = File.ReadAllLines("input.txt");  
 int W = int.Parse(lines[0].Split()[0]);  
 int n = int.Parse(lines[0].Split()[1]);  
  
 int[] values = new int[n];  
 int[] weights = new int[n];  
  
 for (int i = 0; i < n; i++)  
 {  
 string[] item = lines[i + 1].Split();  
 values[i] = int.Parse(item[0]);  
 weights[i] = int.Parse(item[1]);  
 }  
  
 int result = Knapsack(W, weights, values, n);  
 File.WriteAllText("output.txt", result.ToString());  
 }  
}

**9.4.2 Приклад роботи**

На рисунках 9.1 і 9.2 показані приклади вхідного та вихідного файлу:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 9.1 Приклад вхідного файлу

Изображение выглядит как текст, снимок экрана, Шрифт, логотип

Автоматически созданное описание

Рис. 9.2 Приклад вихідного файлу

**9.5 Висновок**

В ході виконання цієї лабораторної роботи, я зрозуміла принцип роботи та використання на практиці динамічного програмування. Побачила, що цей метод розробки алгоритмів доволі потужний для розв’язання задач.