

# Sprawozdanie 10: Programowanie dynamiczne: problem plecakowy

Dmytro Sakharskyi - L02 - 31259

19.06.2025

Metody Programowania

Informatyka 1 rok. 2 semestr

## Wstęp

Celem niniejszego laboratorium było zapoznanie się z różnymi metodami rozwiązywania problemu plecakowego – zarówno w wersji dyskretnej, jak i ciągłej. Zaimplementowano trzy podejścia:

- **naiwne przeszukiwanie wszystkich konfiguracji**,
- **algorytm zachłanny** dla problemu plecakowego ciągłego,
- **programowanie dynamiczne** dla problemu dyskretnego.

Dodatkowo porównano uzyskane wyniki z przykładowymi zestawami wejściowymi (Rysunki 1 i 2) oraz przeprowadzono analizę skuteczności i poprawności poszczególnych algorytmów.

## Zadanie 1

W tym zadaniu zaimplementowano **naiwne rozwiązanie** dyskretnego problemu plecakowego (0-1 knapsack). Algorytm przegląda **wszystkie możliwe konfiguracje** przedmiotów ( $2^n$  możliwości) i wybiera najlepszą spośród nich.

```
int main() {  
    vector<Item> items = {  
        {60, 10, "Przedmiot 1"},  
        {100, 20, "Przedmiot 2"},  
        {120, 30, "Przedmiot 3"}  
    };  
}
```

*Kod na 3 przedmioty.*

## Wynik pracy:

### 1d. 3 przedmioty:

```
--- Wszystkie konfiguracje ---  
  
Konfiguracja: (brak) | Waga: 0 kg, Wartosc: 0 zl  
Konfiguracja: Przedmiot 1 | Waga: 10 kg, Wartosc: 60 zl  
Konfiguracja: Przedmiot 2 | Waga: 20 kg, Wartosc: 100 zl  
Konfiguracja: Przedmiot 1 Przedmiot 2 | Waga: 30 kg, Wartosc: 160 zl  
Konfiguracja: Przedmiot 3 | Waga: 30 kg, Wartosc: 120 zl  
Konfiguracja: Przedmiot 1 Przedmiot 3 | Waga: 40 kg, Wartosc: 180 zl  
Konfiguracja: Przedmiot 2 Przedmiot 3 | Waga: 50 kg, Wartosc: 220 zl  
Konfiguracja: Przedmiot 1 Przedmiot 2 Przedmiot 3 | Waga: 60 kg, Wartosc: 280 zl  
  
--- Rozwiazanie optymalne ---  
Wartosc maksymalna: 220 zl  
Przedmioty: Przedmiot 2 Przedmiot 3 | Waga: 50 kg
```

### 1e. 5 przedmioty:

```

Konfiguracja: Przedmiot 2 Przedmiot 3 Przedmiot 4 | Waga: 65 kg, Wartosc: 110 zl
Konfiguracja: Przedmiot 1 Przedmiot 2 Przedmiot 3 Przedmiot 4 | Waga: 72 kg, Wartosc: 170 zl
Konfiguracja: Przedmiot 5 | Waga: 22 kg, Wartosc: 110 zl
Konfiguracja: Przedmiot 1 Przedmiot 5 | Waga: 32 kg, Wartosc: 170 zl
Konfiguracja: Przedmiot 2 Przedmiot 5 | Waga: 42 kg, Wartosc: 210 zl
Konfiguracja: Przedmiot 1 Przedmiot 2 Przedmiot 5 | Waga: 52 kg, Wartosc: 230 zl
Konfiguracja: Przedmiot 3 Przedmiot 5 | Waga: 52 kg, Wartosc: 230 zl
Konfiguracja: Przedmiot 1 Przedmiot 3 Przedmiot 5 | Waga: 62 kg, Wartosc: 270 zl
Konfiguracja: Przedmiot 2 Przedmiot 3 Przedmiot 5 | Waga: 72 kg, Wartosc: 310 zl
Konfiguracja: Przedmiot 1 Przedmiot 2 Przedmiot 3 Przedmiot 5 | Waga: 82 kg, Wartosc: 350 zl
Konfiguracja: Przedmiot 4 Przedmiot 5 | Waga: 37 kg, Wartosc: 190 zl
Konfiguracja: Przedmiot 1 Przedmiot 4 Przedmiot 5 | Waga: 47 kg, Wartosc: 250 zl
Konfiguracja: Przedmiot 2 Przedmiot 4 Przedmiot 5 | Waga: 57 kg, Wartosc: 310 zl
Konfiguracja: Przedmiot 1 Przedmiot 2 Przedmiot 4 Przedmiot 5 | Waga: 67 kg, Wartosc: 370 zl
Konfiguracja: Przedmiot 3 Przedmiot 4 Przedmiot 5 | Waga: 67 kg, Wartosc: 370 zl
Konfiguracja: Przedmiot 1 Przedmiot 3 Przedmiot 4 Przedmiot 5 | Waga: 77 kg, Wartosc: 430 zl
Konfiguracja: Przedmiot 2 Przedmiot 3 Przedmiot 4 Przedmiot 5 | Waga: 87 kg, Wartosc: 490 zl
Konfiguracja: Przedmiot 1 Przedmiot 2 Przedmiot 3 Przedmiot 4 Przedmiot 5 | Waga: 97 kg, Wartosc: 550 zl

```

--- Rozwiazanie optymalne ---

Wartosc maksymalna: 250 zl

Przedmioty: Przedmiot 1 Przedmiot 4 Przedmiot 5 | Waga: 47 kg

## 1f. 10 przedmioty:

```

Przedmiot 10 | Waga: 192 kg, Wartosc: 980 zl
Konfiguracja: Przedmiot 2 Przedmiot 3 Przedmiot 4 Przedmiot 5 Przedmiot 6 Przedmiot 10 | Waga: 202 kg, Wartosc: 1020 zl
Konfiguracja: Przedmiot 1 Przedmiot 2 Przedmiot 3 Przedmiot 4 Przedmiot 5 Przedmiot 6 Przedmiot 10 | Waga: 212 kg, Wartosc: 1080 zl

```

--- Rozwiazanie optymalne ---

Wartosc maksymalna: 285 zl

Przedmioty: Przedmiot 1 Przedmiot 2 Przedmiot 4 Przedmiot 9 | Waga: 50 kg

## Zadanie 2

Liczba wszystkich możliwych konfiguracji w naiwnym rozwiązaniu problemu plecakowego wynosi  $2^n$ , gdzie  $n$  to liczba przedmiotów.

### Przykładowo:

- Dla **3** przedmiotów:  $2^3 = 8$
- Dla **5** przedmiotów:  $2^5 = 32$
- Dla **10** przedmiotów:  $2^{10} = 1024$

## Zadanie 3

3e/3f

```
--- Przedmioty w kolejności zachłannej ---  
Przedmiot 1: 10 kg, 60 zł, 6.00 zł/kg  
Przedmiot 2: 20 kg, 100 zł, 5.00 zł/kg  
Przedmiot 3: 30 kg, 120 zł, 4.00 zł/kg  
  
--- Przedmioty wybrane do plecaka ---  
Przedmiot 1 (100%)  
Przedmiot 2 (100%)  
Przedmiot 3 (66.67%)  
  
--- Wynik ---  
Łączna waga: 50.00 kg  
Łączna wartość: 240.00 zł
```

Algorytm zachłanny wybrał przedmioty w kolejności malejącej wartości jednostkowej.

W plecaku znalazły się:

- **Przedmiot 1 (100%)**
- **Przedmiot 2 (100%)**
- **Przedmiot 3 (66,67%)**

Daje to łączną wagę **50 kg** oraz wartość **240,00 zł**.

Wynik zgadza się z rozwiązaniem przedstawionym na **(Rysunek 3)** — pełna zgodność.

## 5 przedmioty:

```
--- Przedmioty w kolejności zachłannej ---
Przedmiot 1: 10 kg, 60 zł, 6.00 zł/kg
Przedmiot 2: 20 kg, 100 zł, 5.00 zł/kg
Przedmiot 5: 17 kg, 80 zł, 4.71 zł/kg
Przedmiot 3: 30 kg, 120 zł, 4.00 zł/kg
Przedmiot 4: 5 kg, 20 zł, 4.00 zł/kg

--- Przedmioty wybrane do plecaka ---
Przedmiot 1 (100%)
Przedmiot 2 (100%)
Przedmiot 5 (100%)
Przedmiot 3 (10.00%)

--- Wynik ---
Łączna waga: 50.00 kg
Łączna wartość: 252.00 zł
```

## 10 przedmioty:

```
--- Przedmioty w kolejności zachłannej ---
Przedmiot 1: 10 kg, 60 zł, 6.00 zł/kg
Przedmiot 2: 20 kg, 100 zł, 5.00 zł/kg
Przedmiot 5: 17 kg, 80 zł, 4.71 zł/kg
Przedmiot 6: 23 kg, 100 zł, 4.35 zł/kg
Przedmiot 3: 30 kg, 120 zł, 4.00 zł/kg
Przedmiot 4: 5 kg, 20 zł, 4.00 zł/kg
Przedmiot 8: 38 kg, 150 zł, 3.95 zł/kg
Przedmiot 7: 37 kg, 130 zł, 3.51 zł/kg
Przedmiot 9: 76 kg, 200 zł, 2.63 zł/kg
Przedmiot 10: 120 kg, 300 zł, 2.50 zł/kg

--- Przedmioty wybrane do plecaka ---
Przedmiot 1 (100%)
Przedmiot 2 (100%)
Przedmiot 5 (100%)
Przedmiot 6 (13.04%)

--- Wynik ---
Łączna waga: 50.00 kg
Łączna wartość: 253.04 zł
```

## Zadanie 4

Kod dla 4a+4b:

```

struct Przedmiot {
    int waga;
    int wartosc;
};

void solveKnapsack(const vector<Przedmiot>& przedmioty, int capacity) {
    int n = przedmioty.size();
    vector<vector<int>> dp(n + 1, vector<int>(capacity + 1, 0));

    for (int i = 1; i <= n; ++i) {
        for (int w = 0; w <= capacity; ++w) {
            dp[i][w] = dp[i - 1][w];
            if (przedmioty[i - 1].waga <= w) {
                dp[i][w] = max(dp[i][w],
                    przedmioty[i - 1].wartosc + dp[i - 1][w - przedmioty[i - 1].waga]);
            }
        }
    }
}

```

```

cout << "--- Rozwiazanie dynamiczne ---\n";
cout << "Maksymalna wartosc: " << dp[n][capacity] << " zl\n";

int w = capacity;
int totalWeight = 0;
cout << "Wybrane przedmioty:\n";
for (int i = n; i > 0; --i) {
    if (dp[i][w] != dp[i - 1][w]) {
        cout << "    Przedmiot " << i
            << ": " << przedmioty[i - 1].waga << " kg, "
            << przedmioty[i - 1].wartosc << " zl\n";
        totalWeight += przedmioty[i - 1].waga;
        w -= przedmioty[i - 1].waga;
    }
}

cout << "Laczna waga: " << totalWeight << " kg\n";
}

int main() {
    vector<Przedmiot> przedmioty = {
        {10, 60}, {20, 100}, {30, 120}
    };
    int capacity = 50;

    solveKnapsack(przedmioty, capacity);
    return 0;
}

```

## Wynik pracy:

### 4c. 3 przedmioty:

```
--- Rozwiazanie dynamiczne ---  
Maksymalna wartosc: 220 zł  
Wybrane przedmioty:  
  Przedmiot 3: 30 kg, 120 zł  
  Przedmiot 2: 20 kg, 100 zł  
Laczna waga: 50 kg
```

Dla zestawu z Rysunku 1 algorytm dynamiczny zwrócił maksymalną wartość **220 zł** oraz łączną wagę **50 kg**.

Wybrane przedmioty to **Przedmiot 3 i Przedmiot 2**, czyli dokładnie jak na **Rysunku 2. Wynik potwierdza poprawność działania algorytmu.**

### 5 przedmioty:

```
--- Rozwiazanie dynamiczne ---  
Maksymalna wartosc: 230 zł  
Wybrane przedmioty:  
  Przedmiot 5: 36 kg, 170 zł  
  Przedmiot 1: 10 kg, 60 zł  
Laczna waga: 46 kg
```

### 10 przedmioty:

```
--- Rozwiązanie dynamiczne ---  
Maksymalna wartosc: 282 zł  
Wybrane przedmioty:  
  Przedmiot 10: 12 kg, 77 zł  
  Przedmiot 6: 8 kg, 45 zł  
  Przedmiot 2: 20 kg, 100 zł  
  Przedmiot 1: 10 kg, 60 zł  
Łączna waga: 50 kg
```

## Wnioski

**Naiwne podejście** pozwala znaleźć rozwiązanie optymalne, ale jest bardzo nieefektywne przy większej liczbie przedmiotów ze względu na wykładniczy wzrost liczby konfiguracji.

**Algorytm zachłanny** sprawdza się dobrze w problemie plecakowym ciągłym – daje szybkie i zadowalające wyniki, choć nie zawsze optymalne.

**Programowanie dynamiczne** daje gwarancję znalezienia rozwiązania optymalnego w problemie dyskretnym, przy akceptowalnej złożoności obliczeniowej.

Otrzymane wyniki dla wszystkich metod zgadzają się z oczekiwanymi wartościami z Rysunku 2, co potwierdza poprawność implementacji.