Knapsack Problem – Pakowanie Plecaka

I. Wprowadzenie

Problem plecakowy jest jednym z najczęściej poruszanych problemów optymalizacyjnych. Opracowany przeze mnie algorytm opiera się na podejmowaniu prób umieszczenia w plecaku elementów o różnej wadze, tak aby sumaryczna waga plecaka osiągnęła pewną określoną wartość. Nie ma wymogu, by w plecaku zostały umieszczone wszystkie elementy. Algorytm rekurencyjny po kolei sprawdza różne możliwości dopóki nie odnajdzie tej pożądanej lub też nie skończą się możliwe konfiguracje elementów.

II. Opis interfejsu

Aby uruchomić program należy wpisać w konsoli polecenie *„python Knapsack.py”*, a następnie poczekać na komunikat:

```
E:\UJ FAIS\PYTHON\Projekt>python Knapsack.py
Do you want to add your variables [1] or check proposed ones? [2]
```

Można wtedy wybrać jedną z dwóch opcji, po wybraniu opcji wprowadzenia własnych danych należy nacisnąć na klawiaturze klawisz „1”, w przypadku wybrania opcji proponowanych przykładowych danych należy nacisnąć „2”. Jeśli wprowadzone dane będą niepoprawne program poprosi nas o powtórzenie swojej decyzji:

```
E:\UJ FAIS\PYTHON\Projekt>python Knapsack.py
Do you want to add your variables [1] or check proposed ones? [2] 3
Are you sure your answer was correct? Try again! Do you want to add your variables [1] or check proposed ones? [2]
```

Po wybraniu opcji wprowadzenia własnych danych program kolejno poprosi o wprowadzenie ilości zestawów danych jakie użytkownik chciałby wpisać, pojemność plecaka, ilość pakowanych obiektów oraz ich kolejne wagi:

```
How many of data sets you want to enter? 2
How big is your knapsack? 15
How many objects you want to pack? 5
Add object number 1 3
Add object number 2 2
Add object number 3 10
Add object number 4 4
Add object number 5 4
15 = 3 2 10
How big is your knapsack? 80
How many objects you want to pack? 3
Add object number 1 40
Add object number 2 40
Add object number 3 30
80 = 40 40
```

Jako wynik program poda znalezioną sumę elementów składających się na daną pojemność. W przypadku nieodnalezienia żadnej możliwości wypełnienia plecaka program wyświetli stosowny komunikat:

```
How big is your knapsack? 50
How many objects you want to pack? 2
Add object number 1 1
Add object number 2 1
There is no solution!
```

Po wybraniu opcji gotowych danych program uruchomi się informując nas o danych oraz wyniku przeprowadzanych operacji:

```
Do you want to add your variables [1] or check proposed ones? [2] 2
For data: Size of knapsack = 30 and objects 10, 10, 9, 5, 5
30 = 10 10 5 5
For data: Size of knapsack = 300 and objects 10, 10, 90, 5, 5, 30, 10, 50, 40, 32, 14, 5, 10, 98, 36, 29, 30, 40, 67, 30
300 = 10 10 90 5 5 30 10 50 40 14 36
For data: Size of knapsack = 38 and objects 10, 10, 9, 7, 15
There is no solution!
For data: Size of knapsack = 318 and objects 19, 14, 185, 12, 67, 38, 50, 75, 10, 141, 25, 148, 86, 99, 8, 109, 70, 26, 78, 18, 138, 87, 46, 71
, 60, 38, 33, 83, 191, 23, 4, 53, 172, 12, 146, 160, 186, 15, 136, 108, 130, 104, 67, 125, 83, 21, 117, 129, 71, 45, 68, 71, 19, 137, 101, 134,
 52, 113, 73, 119, 40, 31, 38, 61, 39, 89, 25, 187, 167, 18, 26, 83, 41, 182, 7, 1, 65, 28, 165, 124, 65, 168, 23, 54, 41, 49, 109, 79, 166, 7,
 149, 131, 129, 43, 173, 155, 168, 100, 21, 87, 181, 153, 47, 178, 40, 52, 147, 191, 167, 54, 14, 100, 156, 63, 16, 123, 30, 114, 54, 125, 16,
72, 139, 177, 154, 137, 186, 162, 19, 102, 68, 141, 144, 101, 74, 120, 85, 186, 77, 17, 49, 184, 5, 69, 116, 81, 170, 95, 0, 1, 171, 82, 67, 96
, 171, 118, 95, 114, 151, 6, 186, 110, 30, 118, 79, 153, 155, 146, 12, 171, 157, 43, 23, 12, 59, 79, 157, 61, 155, 20, 59, 4, 123, 6, 70, 179,
137, 79, 169, 107, 158, 100, 183
318 = 19 14 185 12 67 10 4 7
```

Gdzie ostatni z zestawów danych jest każdorazowo losowo generowany.

III. Podsumowanie

Aplikacja jest przyjazna użytkownikom, zwraca jasne i czytelne komunikaty zarówno w przypadku odnalezienia rozwiązania jak i jego braku. Sam algorytm działa bardzo szybko, z wykorzystaniem rekurencji w optymalny sposób odnajdując korzystne wyniki dla każdego zestawu danych. Do korzystania z niej nie są potrzebne żadne zewnętrzne biblioteki, a sama implementacja jest możliwie najbardziej przejrzysta i czytelna.

IV. Literatura

- *Ze zbioru skryptów z wykładów Wydziału Matematyki i Informatyki, przedmiotu Metody Programowania - W06_Rekurencja, dr Jan Rosek*