

# Problem plecakowy

200439

Problem plecakowy to jedno z najczęściej poruszanych zagadnień optymalizacji. Nazwa tego problemu wzięła się z analogii do rzeczywistej sytuacji pakowania plecaka. Należy go zapakować w ten sposób, by pomieścić w nim rzeczy o łącznej największej możliwej wartości. Przy tym nie wolno przekroczyć największej dopuszczalnej wagi plecaka oraz przedmioty nie mogą być rozdzielane na mniejsze części. Istnieje kilka odmian tego problemu, w tym zadaniu został zrealizowany przypadek, gdy do dyspozycji mamy ustaloną ilość różnych rodzajów przedmiotów w nieograniczonej liczbie sztuk. Każdy przedmiot ma swoją nazwę, wagę oraz wartość (cenę). Jak więc ustalić zestaw przedmiotów do spakowania do plecaka, które jak najdokładniej go wypełnią, jednocześnie dając największą wartość zapakowanych przedmiotów?

Jest to problem, którego rozwiązanie wielomianowe nie jest znane - problem NP trudny. Istnieje kilka możliwości rozwiązania tego problemu, takie jak mało efektywna metoda przeglądu zupełnego (bruteforce, metoda siłowa), czy algorytm aproksymacyjny - algorytm zachłanny. Znacznie lepszym sposobem jest metoda programowania dynamicznego. Dwie ostatnie metody zostały zaimplementowane w moim programie.

## 1 Algorytm zachłanny

Algorytm zachłanny jest odwzorowaniem naturalnego podejścia człowieka do pakowania przedmiotów: na początku pakuje się rzeczy najcenniejsze i najlżejsze, tak aby zmieściło się ich możliwie najwięcej. Dlatego w algorytmie zachłannym wylicza się wartości stosunku ceny do wagi przedmiotu. Pozwala to stwierdzić, który przedmiot jest najbardziej wart spakowania. Po obliczeniu tych stosunków, wartości te szereguje się w kolejności malejącej. Następnie wypełnia się plecak przedmiotami, rozpoczynając od tego z największym stosunkiem cena/waga i pakuje się ich największą możliwą ilość. Gdy w plecaku nie ma już miejsca na ten rodzaj przedmiotu, przechodzi się do pakowania kolejnego, z mniejszą wartością cena/waga. Postępuje się tak aż do wypełnienia całego plecaka. Nie jest to jednak algorytm optymalny -

nie gwarantuje najlepszego rozwiązania problemu, co zostanie udowodnione później na przykładzie.

## 2 Algorytm dynamiczny

Lepszą metodą odnalezienia optymalnego rozwiązania problemu plecakowego jest programowanie dynamiczne. Wykorzystuje ono metodę "dziel i zwyciężaj", która rozkłada zagadnienie na szereg mniejszych problemów. Dlatego rozpoczyna się od sytuacji, w której plecak należałoby wypełnić tylko jednym rodzajem przedmiotów. Dodatkowo pojemność maksymalna plecaka również zostaje podzielona i rozszerzana co jednostkę od 1 do wartości maksymalnej. W ten sposób stworzyć można macierz  $P$  upakowania plecaka. Każdy kolejny wiersz takiej tablicy bazuje na poprzedzającym go wierszu, a dopisywane kolejne przedmioty możliwe do włożenia do plecaka są porównywane pod względem optymalności, czy lepszym rozwiązaniem będzie użycie tych nowych przedmiotów, czy wcześniejsze upakowanie już jest optymalne.

Aby przetestować działanie algorytmu zachłannego i dynamicznego stworzyłem dwie sytuacje: najbardziej klasyczna, czyli problem upakowania łupów złodzieja, tak aby zmieścił on jak najwięcej warte przedmioty oraz problem spakowania najbardziej potrzebnych przedmiotów na biwak.

## 3 Przykłady

W przypadku problemu z pakowaniem plecaka na biwak zdefiniowane mamy następujące przedmioty, ich wartości (ceny, w tym przypadku przydatność) oraz wagę:

Nr	Nazwa	Wartość	Waga
1	ubrania	20	14
2	namiot	5	8
3	prowiant	7	5
4	wędka	1	2
5	lornetka	14	10

Tabela 1.

Pojemność plecaka wynosi 24. W tym problemie oba algorytmy odnajdują optymalne rozwiązanie.

Algorytm zachłanny na początku wylicza wartości cena/waga, a następnie sortuje przedmioty malejąco ze względu na ten parametr:

Nr	Nazwa	Wartość	Waga	Cena/waga
1	ubrania	20	14	1,42
2	prowiant	7	5	1,40
3	namiot	5	8	0,63
4	lornetka	6	10	0,60
5	wędka	1	2	0,50

Tabela 2.

Następnie wypełnia plecak przedmiotami w kolejności od najwyżej stojącego w tabeli powyżej, w tym wypadku to ubrania. Mają wagę równą 14, więc w naszym plecaku o pojemności 24 zmieści się jeden taki przedmiot. Pozostała ilość wolnego miejsca to  $24 - 14 = 10$ . W tym wypadku plecak dopełniany jest przedmiotami będącymi następnymi w tabeli, czyli prowiant. Ma on wagę 5, więc zmieści się podwójna jego ilość. Teraz plecak jest już pełny, a łączna wartość rzeczy, które udało się zmieścić to 34.

Algorytm dynamiczny rozwiązuje problem w inny sposób. Tworzy macierz, w której numer wiersza odpowiada numerowi przedmiotu z tabeli 1 zaś numer kolumny jest kolejną całkowitą pojemnością plecaka od 1 do maksimum równego 24. Tak stworzona macierz wypełniana jest kolejnymi możliwymi upokowaniami, w każdym wierszu dodając nowy rodzaj przedmiotu. W ostatniej kolumnie ostatniego wiersza uzyskamy wartość optymalnego upakowania plecaka. Wygenerowaną taką macierz można zobaczyć po uruchomieniu programu. Wartość obliczona algorytmem dynamicznym jest identyczna z wartością wyliczoną przez algorytm zachłanny.

Nie jest tak jednak zawsze. W drugim przykładzie, ze złodziejem, zauważyć można, że algorytm dynamiczny daje lepsze i bardziej optymalne rozwiązanie problemu niż algorytm zachłanny. Mamy do dyspozycji następujące przedmioty:

Nr	Nazwa	Wartość	Waga
1	naszyjnik	5000	7
2	pierścione	2100	3
3	sztabka złota	25000	11
4	diament	2000	2

Tabela 3.

Plecak ma pojemność 20. Algorytm zachłanny pozwala na zapakowanie 4 diamentów o wartości 2000 i sztabkę złota o wartości 25000. Łącznie daje to wartość 33000. Zostaje jeszcze wolna przestrzeń o wartości 1, lecz nie można jej wypełnić, gdyż żaden z przedmiotów nie ma tak małej wagi. Algorytm dynamiczny zaś pozwala na zapakowanie 3 diamentów o wartości 2000, sztabki złota o wartości 25000 oraz pierścionka o wartości 2100. Plecak w tym wypadku zostaje całkowicie wypełniony i wartość przedmiotów zapakowanych wynosi 33100, co jest wartością o 100 większą, z czego złodziej na pewno byłby bardzo zadowolony.

## 4 Podsumowanie

Algorytm zachłanny jest bardzo intuicyjny i prosty. Daje on jednak często wyniki nieoptymalne. Lepszym algorytmem rozwiązującym problem plecakowy jest algorytm dynamiczny. Mimo, że problem plecakowy jest problemem NP zupełnym, jego rozwiązanie jest generowane przez ten algorytm w bardzo dobry sposób.