Dyskretny problem plecakowy

Wyobraźmy sobie następujący problem optymalizacyjny (zwany problemem plecakowym). Mamy zbiór przedmiotów każdy o określonej wadze oraz wartości. Mamy też plecak, który może pomieścić przedmioty o określonej wadze sumarycznej (bo inaczej się urwie). Zagadnienie polega na tym, by wybrać do plecaka takie przedmioty, by ich sumaryczna wartość była jak największa.

Jest to dość popularne zagadnienie optymalizacyjne. W dalszych rozważaniach przyjmujemy, że zarówno rozmiar plecaka jak i wagi (a także wartości) przedmiotów są liczbami naturalnymi.

Proste ale niekoniecznie optymalne metody rozwiązywania: Algorytm zachłanny

Ogólnie algorytm zachłanny szuka najbardziej optymalnego rozwiązania w danej chwili (w każdym kroku). W tym przypadku sortujemy rzeczy po cenie malejąco i dodajemy do plecaka kolejno rzeczy o największej wartości aż dodanie kolejnej rzeczy przekroczyłoby dopuszczalną masę sumaryczną. To nie zawsze jest najlepsze rozwiązanie. Załóżmy że mamy trzy rzeczy i plecak o pojemności do 50kg.

1) Waga 20 Wartość 10
2) Waga 40 Wartość 15
3) Waga 25 Wartość 8

Opisany wyżej algorytm zachłanny doda do plecaka rzecz drugą o wartości 15 (i masie 40). Więcej już się nie zmieści. Optymalne rozwiązanie to spakować rzecz 1 i 3 co da łączną wartość 18 i wagę 45 (a więc mieszczącą się w limicie).

Inna wersja: policzyć dla każdej rzeczy stosunek wartości do masy i pakować rzeczy wg wartości tego stosunku. To też nie zawsze jest optymalne rozwiązanie.

Właściwe rozwiązanie to

Algorytm dynamiczny

Jego pseudokod można znaleźć (pod nazwą Knapsack Problem) np. w angielskie wersji Wikipedii:

https://en.wikipedia.org/wiki/Knapsack_problem

Problem rozwiązuje się tak.

1) Tworzymy macierz X o tylu kolumnach ile wynosi dopuszczalna masa jaką pomieści plecak +1 (np. dopuszczalna masa to 15kg to macierz ma 16 kolumn). Liczbę kolumn określamy jako N. Liczba wierszy w macierzy to liczba przedmiotów +1 (dodanie dodatkowego wiersza ułatwi dalsze obliczenia, gdyż będzie można bezpiecznie odwoływać się do wiersza wcześniejszego).
2) W pierwszym wierszu (indeks zerowy) dodajemy same zera
3) Każdy kolejny wiersz odpowiada za dodawanie poszczególnych dostępnych przedmiotów. Załóżmy że i-ty przedmiot ma wagę (masę) W_i oraz wartość (cenę) C_i . Kolumny indeksujemy literą j (a wiersze literą i).
4) Gdy dodajemy pierwszy przedmiot (wiersz numer 1) to dla kolumn o $W_i > j$ dajemy 0 a dla $W_i <= j$ dajemy C_i. (innymi słowy, przedmiot możemy włożyć do plecaka, tylko wtedy gdy plecak „wytrzymuje” ciężar większy lub równy W_i. Kolumny określają właśnie maksymalny ciężar jaki wytrzyma plecak.

5) Teraz sprawdzamy kolejne przedmioty. W kolejnych wierszach (indeksowanych przez i) w kolumnie j-tej

   a.) Sprawdzamy czy $W_i>j$, jeśli tak to X[i,j]= X[i-1,j] (innymi słowy do plecaka z kolumny j-tej przedmiot się nie zmieści -nawet jeśli plecak jest pusty- , przepisujemy zatem „wartość" zawartości plecaka z komórki leżącej o jedno pole wyżej).

   b. Jeśli $W_i<=j$ dajemy maksimum z (X[i-1,j] oraz X[i-1,j-W_i]+c_i) Oznacza to co następuje:

      Podpunkt a) Jeśli kolejny przedmiot jest cięższy od maksymalnej masy jaką można trzymać w plecaku, to **nie możemy go dodać**. Wtedy optymalne rozwiązanie to rozwiązanie dla plecaka bez tego przedmiotu (komórka o jeden wiersz wyżej)

      Podpunkt b) Jeśli przedmiot „mieści" się w plecaku (masowo), to trzeba sprawdzić, czy jego dodanie się opłaca. W tym celu szukamy maksimum z dwóch liczb, wartości „optymalnego" plecaka bez przedmiotu (komórka nad aktualną komórką czyli X[i-1,j]), oraz wartości plecaka z komórki X[i-1,j-W_i] powiększonej o wartość C_i

6) Finalnie gdy przeanalizujemy wszystkie dostępne przedmioty optymalna „wartość" plecaka jest w prawym dolnym rogu tabeli X.

Przykład. Maksymalna wartość zawartości plecaka o „rozmiarze" 20kg wypełnionego przedmiotami z pliku PlecakTest.txt to 590.