Implementacja algorytmu rozwiązującego problem plecakowy

Arkadiusz Cyktor 200367

28 maja 2014

Problem plecakowy można opisać w skrócie, jako dobór takich elementów zbioru, abych ich wartość była jak największa, przy jednoczesnym utrzymaniu ich łącznej wagi poniżej określonego poziomu.

Zagadnienie to można rozwiązać na kilka sposobów, ja w swoim programie zdecydowałem się na **programowanie dynamiczne**. Takie podejście do problemu polega na rzobiciu go na kilka mniejszych podproblemów (potencjalnie łatwiejszych do rozwiązania), a następnie na wykorzystaniu ich rozwiązań do uzyskania nteresującego nas wyniku.
W przypdaku problemu plecakowego polega to na rozwiązaniu szeregu zadań, w których rozmiar "plecaka"jest znacznie mniejszy od zadanego.

W swoim programie zastosowałem dwuwymiarową tablicę o *N+1* wierszach (gdzie N jest zadaną wielkością "plecaka"), które przedstawiają zwiększający się (od 0 do N) rozmiar problemu, oraz o *X+1* kolumnach, przy czym każda z nich odpowiada jednemu przedmiotowi z zadanego zbioru. Zasada działania algorytmu jest bardzo prosta - dla każdego wiersza obliczane jest optymalne wypełnienie "plecakaź uwzględnieniem kolejno każdego elementu zbioru, następnie wynik zapisywany jest (przy użyciu klasy *Pole_tablicy*) w polu na przecięciu danego wiersza i kolumny. Jeśli któryś z elementów przekracza aktualny rozmiar "plecaka", to w dane pole zostaje przepisana wartość pola poprzedniego. Natomiast w wypadku, gdy element można z powodzeniem umieścić w "plecakuąlgorytm cofa się do pola (w tej samej kolumnie), dla którego pojemność plecaka była mniejsza o wagę aktualnie rozpatrywanego elementu i dodaje do jego łącznej wartości wartość obecnego elementu. W ten sposób odwołujemy się do innego najlepszego upakowania elementów, które zostało już wcześniej policzone. Tak uzyskana konfiguracja porównywana jest z poprzednią wartością z wiersza i do aktualnie przetwarzanego pola zapisywana jest ta o większej łącznej wartości upakowanych elementów.
Przechodząc w ten sposób wszystkie wiersze i kolumny uzyskamy w ostatnim polu optymalne rozwiązanie głównego problemu.
Ponieważ liczba wpisów w tablicy wyników wynosi w*k, (gdzie w to liczba wierszy, a k - kolumn) to złożonością obliczeniową takiego algorytmu jest O(wk).

W programie zostały wprowadzone trzy klasy *Przedmiot*, *Pole_tablicy* i *Glowna*. Pierwsza z nich odpowiada za modelowanie przedmiotów, które mogą być umieszczane w "plecaku- zawiera informacje o ich wadze i wartości, a także indeks, który pozwala na ich identyfikację. Druga klasa odpowiada za realizację pól wspomniajej wcześniej tablicy - znajduje się w niej wektor przechowujący indeksy przedmiotów składających się na optymalne rozwiązanie danego problemu, a także pole typu int przechowujące sumaryczną wartość wspomnianych przedmiotów. Ostatnia klasa przechowuje tablicę rozwiązań oraz dwie globalne zmienne - przechowujące informacje o wielkości "plecakaóraz ilości elementów.