

Lipski Kamil Krzysztof

Analiza algorytmów

prowadzący: dr inż. Tomasz Trzeciński

zadanie 12.

Treść zadania

Do walizki o ograniczonej pojemności C chcemy załadować przedmioty o jak największej wartości, mając jednak na uwadze, że każdy z nich zajmuje pewną objętość. Mając n przedmiotów wraz z n -elementową tablicą odpowiadających im wartości $\{p_i\}$ oraz objętości $\{c_i\}$, znajdź zestaw rzeczy mieszczących się w walizce o największej sumarycznej wartości. *Uwaga:* Możemy dobierać maksymalnie m przedmiotów tego samego typu (tzn. tej samej wartości i tej samej objętości). Oceń złożoność algorytmu.

Interpretacja problemu

Zagadnienie przedstawione w treści zadania sprowadza się do dyskretnego problemu plecakowego z dodatkowym parametrem m , który wprowadza ograniczenie dotyczące maksymalnej ilości przedmiotów tego samego typu, jaką możemy umieścić w walizce.

Założenia

- z oczywistych względów każda z wielkości wymienionych w treści zadania powinna być liczbą nieujemną – zagadnienie do rozwiązania jest reprezentacją realnego problemu
- w zadaniu będę posługiwał się liczbami całkowitymi – na tym założeniu opiera się jeden z pomysłów zastosowanych w rozwiązaniu, a mianowicie – zliczanie kubelkowe; ponadto zmienne n i m muszą być liczbami całkowitymi, jako że do dyspozycji mamy jedynie **całe** przedmioty, a nie ich części
- pojemność C jest liczbą dodatnią – w przeciwnym przypadku nie będzie istniała możliwość umieszczenia przedmiotów w walizce
- parametr m również jest liczbą dodatnią – w przeciwnym przypadku żaden z przedmiotów nie będzie mógł być umieszczony w walizce (szczególny przypadek: $m = 1$ – inna odmiana problemu plecakowego)
- ilość przedmiotów n jest liczbą dodatnią – aby rozpatrywanie problemu miało sens, muszą być dostępne jakiekolwiek obiekty do spakowania

- wartości przedmiotów p_i są liczbami nieujemnymi – w przypadku, gdy wszystkie z nich będą miały wartości zerowe, problem sprowadzi się do umieszczenia w walizce jak największej ilości przedmiotów (uwzględnianie samych objętości obiektów)
- objętości przedmiotów c_i są liczbami dodatnimi – każdy obiekt ma fizyczne parametry, które w przypadku tego zagadnienia reprezentuje objętość
- jako że zadanie jest reprezentacją realnego problemu, to każdy przedmiot może być umieszczony w walizce tylko raz; dostępne elementy mogą się jednak powtarzać - generując n przedmiotów dopuszczam po prostu możliwość tworzenia obiektów o powtarzających się zestawach parametrów; dlatego też istnieje możliwość, że parametr m nie będzie w danym przypadku ograniczeniem – chociażby w sytuacji, gdy ilość przedmiotów będących tego samego rodzaju dla każdego z typów obiektów będzie mniejsza od tego parametru

Opis rozwiązania (algorytmu)

Na potrzeby zadania zostanie zdefiniowana nowa struktura danych *dataCollection*, przechowująca dane odpowiadające danej instancji problemu.

Jednym z założeń zadania jest, że możemy dobrać maksymalnie m przedmiotów tego samego typu. Oznacza to, że jeśli mamy więcej niż m elementów pewnego rodzaju, to obiekty nadmiarowe nie będą uwzględniane w rozwiązaniu. Rozsądne jest więc, aby pozbyć się ich przed wykonaniem właściwej części algorytmu. W tym celu zamierzam skorzystać z algorytmu zliczania kubelkowego. W utworzonej na jego potrzeby strukturze danych (wektor wektorów), w odpowiadających danemu typowi przedmiotu miejscach, będę zliczał wystąpienia obiektu tego rodzaju (wektor „nadrzędny” odpowiadał będzie objętościom przedmiotów c_i , a wektory „podrzędne” – wartościom przedmiotów p_i). Po przekroczeniu dozwolonej liczby wystąpień, przedmioty nadmiarowe dla danego typu obiektu będą na bieżąco usuwane.

Po początkowym przetworzeniu danych, przystąpię do właściwej części zadania – do rozwiązania głównego problemu planuję użyć programowania dynamicznego. Algorytm ma zmaksymalizować wartość elementów przy zachowaniu sumy ich wagi mniejszej bądź równej C . Niech $A(i, j)$ będzie największą możliwą wartością, która może być otrzymana przy założeniu wagi mniejszej bądź równej j i wykorzystaniu pierwszych i elementów. $A(n, C)$ jest rozwiązaniem problemu. Funkcja $A(i, j)$ definiowana jest rekurencyjnie:

- $A(0, j) = 0$
- $A(i, 0) = 0$
- $A(i, j) = A(i - 1, j)$, jeśli $c_i > j$
- $A(i, j) = \max(A(i - 1, j), p_i + A(i - 1, j - c_i))$, jeśli $c_i \leq j$

Złożoność obliczeniowa algorytmu

Zliczanie kubełkowe daje złożoność liniową $O(n)$. W programowaniu dynamicznym wartości $A(i, j)$ do obliczenia jest $n \cdot C$, złożoność obliczeniowa tego algorytmu wynosi więc $O(nC)$. W sumie daje to więc: $f(n) = n + n \cdot C$, czyli $O(nC)$.