Problem plecakowy (Knapsack problem - engl.)

Mamy:
1) „plecak" ma maksymalną wagę (albo pojemność) W_{max}
2) n przedmiotów. Dla każdego przedmiotu wiadomo jego cenę c_i, jego wagę w_i i jego ilość maksymalną m_i, $i=1,2,\dots,n$.

Ogólny

Trzeba znaleźć taki wektor
$(k_1, k_2, \dots, k_n)$,
gdzie k_i - ilość i-tego przedmiotu
Ograniczenia
$$\begin{cases} 0 \le k_i \le m_i \\ \sum_{i=1}^{n} k_i w_i \le W_{max} \end{cases}$$
Zadanie $C = \sum_{i=1}^{n} k_i c_i \longrightarrow \max$

Decyzyjny

Trzeba znaleźć taki wektor
$(k_1, k_2, \dots, k_n)$,
gdzie $k_i = 0$ albo $k_i = 1$, $i = 1,2,\dots,n$.
Ograniczenie
$$\sum_{i=1}^{n} k_i w_i \le W_{max}$$
Zadanie $C = \sum_{i=1}^{n} k_i c_i \longrightarrow \max$

Algorytm zachłanny

1 sposób Posortować przedmioty wg. cen

$$C_1 \geq C_2 \geq \ldots \geq C_n$$

2 sposób Posortować przedmioty wg. wag

$$w_1 \leq w_2 \leq \ldots \leq w_n$$

3 sposób Posortować przedmioty wg. opłacalności

$$\frac{C_1}{w_1} \geq \frac{C_2}{w_2} \geq \ldots \geq \frac{C_n}{w_n}$$

Przykład $W_{max} = 23$

P_i	c_i	w_i	c_i/w_i
P_1	6	6	1
P_2	4	2	2
P_3	5	3	1,7
P_4	7	2	3,5
P_5	10	3	3,3
P_6	2	1	2

1 sposób $P_5, P_4, P_1, P_3, P_2, P_6$

7 przedm. N5 $(7 \cdot 3 = 21\,kg)$

$23 - 21 = 2\,kg$ zostało miejsca

1 przedm N4 $(1 \cdot 2 = 2\,kg)$

$2 - 2 = 0\,kg$ zostało miejsca

$C = 7 \cdot 10 + 1 \cdot 7 = 77$ (zł) wektor $(0,0,0,1,7,0)$

Ale!!!

10 przedm. N4 $(10 \cdot 2 = 20\,kg)$

$23 - 20 = 3\,kg$ zost. miejsca

1 przedm. N5 $(1 \cdot 3 = 3\,kg)$

$3 - 3 = 0\,kg$ zost. miejsca

$C = 10 \cdot 7 + 1 \cdot 10 = 80$ (zł)

wektor $(0,0,0,10,1,0)$

2 sposób $P_6, P_4, P_2, P_5, P_3, P_1$

23 przedm. N6 $(23 \cdot 1 = 23\,kg)$

$23 - 23 = 0$ zostało miejsca

$C = 23 \cdot 2 = 46$ (zł) wektor $(0,0,0,0,0,23)$

3 sposób $P_4, P_5, P_2, P_6, P_3, P_1$

11 przedm. N4 $(11 \cdot 2 = 22\,kg)$

$23 - 22 = 1\,kg$ - zost. miejsca

1 przedm. N6 $(1 \cdot 1 = 1\,kg)$

$1 - 1 = 0$ zost. miejsca

$C = 11 \cdot 7 + 1 \cdot 2 = 79$ (zł)

wektor $(0,0,0,11,0,1)$

Algorytm (zachłanny)

1) Posortować przedmioty wg. jednego z 3 sposobów
2) Dla wszystkich przedmiotów (for $i = 1$ to n)
   - 2a) znaleźć ilość i-tego przedmiotu tak, żeby $0 \le k_i \le m_i$ i $k_i w_i \le W_{max}$
   - 2b) $W_{max} = W_{max} - k_i w_i$ (ile miejsca zostało?)
   - 2c) $i = i + 1$ i wrócić na 2a)
3) $C = \sum_{i=1}^{n} k_i c_i$ - cena

Wyniki

$(k_1, k_2, \dots, k_n)$ - ilości przedmiotów

C - cena

Programowanie dynamiczne

Przykład $W_{max} = 10$ Mamy 6 przedmiotów, które opisane w czarnej części tablicy

P_i	c_i	w_i	0	1	2	3	4	5	6	7	8	9	10
			0	0	0	0	0	0	0	0	0	0	0
P_1	75	7	0	0	0	0	0	0	0	75	75	75	75
P_2	150	8	0	0	0	0	0	0	0	75	150	150	150
P_3	250	6	0	0	0	0	0	0	250	250	250	250	250
P_4	35	4	0	0	0	0	35	35	250	250	250	250	285
P_5	10	3	0	0	0	10	35	35	250	250	250	260	285
P_6	100	9	0	0	0	10	35	35	250	250	250	260	285

285 – to maksymalna cena

$(0, 0, 1, 1, 0, 0)$ – wektor

$$260 = \max\{250, 250 + 10\}$$

Niech T_{ij} - elementy tablicy, $i = 0, 1, 2, \dots, n$, $j = 0, 1, 2, \dots, W_{max}$

Realizacja

1) $T_{ij} = 0$ dla $i=0$ i $T_{ij} = 0$ dla $j=0$

2) Dla wszystkich przedmiotów (for $i = 1$ to n)

Dla wszystkich wag maksymalnych (for $j = 1$ to W_{max})

Jeśli $w_i \leq j$ wtedy $T_{ij} = \max\{T_{i-1,j}; c_i + T_{i-1,j-w_i}\}$

inaczej $T_{ij} = T_{i-1,j}$

Żeby znaleźć jakie przedmioty zostały wybrane, trzeba zbudować dodatkową tablicę K_{ij}, $i = 0, 1, \dots, n$, $j = 0, 1, \dots, W_{max}$

1) $K_{ij} = 0$ dla $i=0$, $K_{ij} = 0$ dla $j=0$

2) Dla wszystkich przedmiotów

Dla wszystkich wag maksymalnych

Jeśli $(w_i \leq j)$ i $(T_{i-1,j} < c_i + T_{i-1,j-w_i})$ wtedy $K_{ij} = 1$

inaczej $K_{ij} = 0$

Żeby wyprintować numery wybranych przedmiotów

1) $j = W_{max}$

2) for $i = n$ downto 1 (licznik i zmniejsza się od n do 1)

jeśli $K_{ij} = 1$

wtedy a) Print(i) (i-ty przedmiot został wybrany)

b) $j = j - w_i$