

Ćwiczenie 6. Algorytmy zachłanne i dynamiczne

- Napisz procedurę **plecak** służąc do rozwiązywania problemu plecakowego algorytmem zachłannym. Procedura powinna umożliwiać wybór strategii działania algorytmu (1,2 lub 3 jak na wykładzie). Do budowy procedury wykorzystaj jedną z zaimplementowanych na wcześniejszych zajęciach metod sortowania. Przetestuj procedurę na przykładach z wykładu.

Dane 1:

j	przedmiot	waga, wj	wartość, cj
1	telefon	1	500
2	PS3	7	800
3	laptop	8	900
4	PS Vita	2	200
5	wino	2	100
6	parasol	2	100
7	gitara 1	6	500
8	trampki	4	100

Waga plecaka – 14 kg

Dane 2:

i	P _i	c _i [zł]	m _i [jm]
1	Koszula flanelowa	75	7
2	Spodnie dżinsowe	150	8
3	Sweter	250	6
4	Czapka baseballowa	35	4
5	Kapelówki	10	3
6	Obuwie sportowe	100	9

Waga plecaka - 10 jm

- Dostosuj procedury lub reprezentację danych tak aby algorytmy zachłanne można było zastosować do problemu rozcinania pręta.

Długość	Cena
1	1
2	5
3	8
4	9
5	10
6	17
7	17
8	20
9	22
10	24

- Zaimplementuj ogólny algorytm dynamiczny poszukiwania rozwiązania problemu optymalizacji, według schematu przedstawionego na wykładzie.

DYNAMIC-GENERAL-KNAPSACK

Dane:

$P_i(m_i, c_i), i = 1, 2, \dots, n;$

gdzie P_i – przedmiot, m_i – masa i-tego przedmiotu, c_i – cena i-tego przedmiotu
 M_{\max} – nośność (pojemność) plecaka.

Wyniki:

Tablica wartości $P_{i,j}$ najlepszych upakowań plecaka o pojemności j rzeczami rodzajów od 1 do i ; dla $i=1, 2, \dots, n$ oraz $j=1, 2, \dots, M_{\max}$.

Tablica $Q_{i,j}$ skojarzona z $P_{i,j}$ rzeczy P_i pakowanych do plecaka w ostatnim ruchu.

- 1) {Ustalenie wartości początkowych tablic P i Q rozszerzonych dla ujednolicenia obliczeń o wiersze i kolumny zerowe.}
Dla $j=1, 2, \dots, M_{\max}$ przypisz $P_{0,j}:=0$, $Q_{0,j}:=0$
Dla $i=1, 2, \dots, n$ przypisz $P_{i,0}:=0$, $Q_{i,0}:=0$.
- 2) Dla kolejnych rzeczy $i=1, 2, \dots, n$ wykonaj krok 3.
- 3) Dla kolejnych pojemności plecaka $j=1, 2, \dots, M_{\max}$ wykonaj krok 4.
- 4) Jeśli $j \geq m_i$ {Czyli pojemność plecaka jest wystarczająca, by pomieścić rzecz i }
oraz $P_{i-1,j} < P_{i,j-m_i} + c_i$ to przypisz $P_{i-1,j} = P_{i,j-m_i} + c_i$ oraz $Q_{i,j} = i$, a w przeciwnym razie pozostaw wartości z poprzedniego wiersza, czyli przypisz $P_{i,j} = P_{i-1,j}$ oraz $Q_{i,j} = Q_{i-1,j}$.

KONIEC

Pseudo kod

```
KnapSack( $v, w, n, W$ )
{
    for ( $w = 0$  to  $W$ )  $V[0, w] = 0$ ;
    for ( $i = 1$  to  $n$ )
        for ( $w = 0$  to  $W$ )
            if ( $w[i] \leq w$ )
                 $V[i, w] = \max\{V[i - 1, w], v[i] + V[i - 1, w - w[i]]\}$ ;
            else
                 $V[i, w] = V[i - 1, w]$ ;
    return  $V[n, W]$ ;
}
```

Pseudo kod z odtworzeniem informacji o zestawie rzeczy zapakowanych do plecaka

```
KnapSack( $v, w, n, W$ )
{
    for ( $w = 0$  to  $W$ )  $V[0, w] = 0$ ;
    for ( $i = 1$  to  $n$ )
        for ( $w = 0$  to  $W$ )
            if ( $(w[i] \leq w)$  and ( $v[i] + V[i - 1, w - w[i]] > V[i - 1, w]$ ))
            {
                 $V[i, w] = v[i] + V[i - 1, w - w[i]]$ ;
                keep[i, w] = 1;
            }
            else
            {
                 $V[i, w] = V[i - 1, w]$ ;
                keep[i, w] = 0;
            }
     $K = W$ ;
    for ( $i = n$  downto 1)
        if (keep[i, K] == 1)
        {
            output i;
             $K = K - w[i]$ ;
        }
    return  $V[n, W]$ ;
}


---


 $K = W$ ;
for ( $i = n$  downto 1)
    if (keep[i, K] == 1)
    {
        output i;
         $K = K - w[i]$ ;
    }
```

Rozwiąż zadanie 1, 2 i porównaj otrzymane wyniki z obu metod.

4. Zaimplementuj algorytm zachłanny rozwiązania *decyzyjnego* problemu plecakowego.
5. Zastosuj funkcję Solver systemu Excel do rozwiązania powyższych zadań.