Algorytmy zachłanne a algorytmy dynamiczne

Algorytm zachłanny znajduje rozwiązanie w wyniku podjęcia szeregu decyzji, z których każda wydaje się **w danym momencie najlepsza**. Oznacza to, że każdy dokonany wybór jest **optymalny lokalnie**. Podejście takie daje nadzieję, że osiągnie się rozwiązanie optymalne globalnie, choć nie zawsze tak jest. Przykładem algorytmu zachłannego jest **wydawanie reszty jak najmniejsza ilością monet**. Strategia zachłanna mogłaby wyglądać następująco:

```
dopóki są jeszcze monety i nie wydano całej reszty:
     1. pobierz największą dostępną monetę,
     2. jeśli dodanie monety powoduje przekroczenie kwoty:
          odrzuć monety o tym nominale
     3. w przeciwnym wypadku:
          dodaj monetę do reszty
```

Algorytm zachłanny nie zawsze gwarantuje otrzymania optymalnego rozwiązania. Jednak dla problemu wydawania reszty w większości stosowanych obecnie systemów monetarnych działa, nie działa np. dla systemu (1, 4, 5) (spróbuj wydać kwotę 8)

Problem plecakowy

Problem – jak zapakować do plecaka o ograniczonej pojemności możliwie najbardziej wartościowe rzeczy?
Dane jest n rzeczy R_1, R_2, R_3, … , R_n
oraz dla każdej z nich: p_i – wartość rzeczy nr i, w_i – ile z pojemności plecaka zajmie rzecz nr i.
a także maksymalna pojemność plecaka W

i	1	2	3	4	5	6	W
p_i	6	4	5	7	10	2	
w_i	6	2	3	2	3	1	23

Matematycznie odpowiada to znalezieniu nieujemnych liczb całkowitych: q_1, q_2 , ..., q_n , dla których całkowita wartość plecaka jest największa i jednocześnie spełniony jest warunek, że pojemność plecaka W nie zostaje przekroczona:

$$w_1 q_1 + w_2 q_2 + \ldots + w_n q_n \leq W$$

Wersja decyzyjna problemu plecakowego polega na założeniu, że wielkości *qi* mogą przyjmować wartości 1- gdy rzecz *i* dokładamy do plecaka lub 0 - gdy jej nie bierzemy (***problem plecakowy 0-1***)

Problem plecakowy - podejście zachłanne:

Przy pakowaniu plecaka można użyć jednej ze strategii:

- wybierać najcenniejsze rzeczy czyli w kolejności nierosnących wartości ***pi ,***
- wybierać rzeczy zajmujące najmniej miejsca, czyli w kolejności niemalejących pojemności ***wi ,***
- wybierać rzeczy najcenniejsze w stosunku do zajmowanego miejsca, czyli w kolejności nierosnących wartości ilorazu ***pi/wi***

Problem plecakowy - podejście dynamiczne:

W podejściu dynamicznym rozwiązujemy wszystkie mniejsze podproblemy, dochodząc stopniowo do rozwiązania naszego problemu, na każdym kroku podejmujemy najlepszą decyzję uwzględniając decyzje wcześniejsze.
Dla problemu plecakowego mniejsze problemy odpowiadają mniejszej liczbie rodzajów pakowanych rzeczy (od 1 do n) i mniejszej pojemności plecaka (od 1 do W) .
Rozwiązanie tworzy taką oto tablicę:

j / i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	0	0	0	0	6	6	6	6	6	6	12	12	12	12	12	12	18	18	18	18	18	18
2	0	4	4	8	8	12	12	16	16	20	20	24	24	28	28	32	32	36	36	40	40	44	44
3	0	4	5	8	9	12	13	16	17	20	21	24	25	28	**29**	32	33	36	37	40	41	44	45
4	0	7	7	14	14	21	21	28	28	35	35	42	42	49	49	56	56	63	63	70	70	77	77
5	0	7	10	14	17	21	24	28	31	35	38	42	45	49	52	56	59	63	66	70	73	77	80
6	2	7	10	14	17	21	24	28	31	35	38	42	45	49	52	56	59	63	66	70	73	77	**80**

i - numeracja kolejnych rzeczy wykorzystywanych przy pakowaniu
j - numeracja kolejnych pojemności plecaka

Dany element tej tablicy rozumiemy jako wartość optymalnego wypełnienia plecaka o pojemności *j* rzeczami, których numery mieszczą się między *1,* a *i, czyli na przykład pogrubione 29 [wiersz3, kolumna15] to wartość najcenniejszego plecaka o pojemności 15 upakowanego rzeczami numer:1,2,3.*

Pola tablicy P są wypełniane wierszami. Wypełnienie pierwszego wiersza jest proste, bo dysponujemy tylko rzeczami nr 1 i możemy ich wziąć 1, 2 lub 3 sztuki zależnie od pojemności plecaka (czyli danej kolumny). Przy wypełnianiu kolejnych wierszy korzystamy z już wypełnionych wierszy, a dokładniej: korzystamy jedynie z ostatniego, jeden powyżej danego, zgodnie z zasadą, że na każdym kroku należy podejmować najlepszą decyzję z uwzględnieniem stanu wynikającego z poprzednich decyzji.

W wierszu drugim chcielibyśmy mieć najcenniejsze upakowania plecaka rzeczami nr 1 i nr 2 wykorzystujących wszystkie pojemności od 1 do W. A więc dla pojemności, w których nie zmieści się rzecz nr 2, jest to wartość z wiersza powyżej, a dla pojemności większych lub równych pojemności rzeczy nr 2, tzn jeśli numer kolumny $j \geq w_2$, podejmujemy decyzję czy cenniejsze jest upakowanie tej pojemności rzeczami nr 1 czy może wyjąć ich tyle by zmieściła się jedna rzecz nr 2 i ją dołożyć? A więc wybieramy większą z dwóch wielkości P[1][j] lub P[2][,j-w_2] + p_2 gdzie:

P[1][j]- wartość najlepszego wypełnienia plecaka o pojemności j rzeczami nr 1 (czyli element w wierszu powyżej),

P[2][,j-w_2] + p_2 - wartość wypełnienia plecaka, składającego się z jednej sztuki rzeczy nr 2 oraz optymalnego wypełnienia reszty przestrzeni w plecaku (czyli j-w_2) rzeczami nr 1 lub 2.

Aby wiedzieć jaki zestaw rzeczy tworzy poszczególne wypełnienie tworzymy tablicę Q skojarzoną z P. Wartością Q[i][j] jest numer rzeczy, która ostatnia trafiła do plecaka o pojemności j, gdy dysponujemy rzeczami o numerach od 1 do i.

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	0	**2**	3	**2**	3	**2**	3	**2**	3	**2**	3	**2**	3	2	**3**	2	3	2	3	2	3	2	3
4	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	0	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5
6	6	**4**	5	**4**	5	**4**	5	**4**	5	**4**	5	**4**	5	**4**	5	**4**	5	**4**	5	**4**	5	4	**5**

Aby znaleźć skład plecaka startujemy z pola Q6,23 (rzecz nr 5) i przenosimy się do kolumny mniejszej o pojemność rzeczy z pozycji Q6,23 (rzecz nr 5 zajmuje 3 jednostki pojemności) i dalej podobnie.

Algorytm:

Dane:

n rzeczy R_1, R_2, R_3, … , R_n (każda w nieograniczonej ilości)
oraz dla każdej z nich:
p_i – wartość rzeczy nr i,
w_i – ile z pojemności plecaka zajmie rzecz nr i.
a także maksymalna pojemność plecaka W

Wynik:

Wartość najcenniejszego plecaka o pojemności W jaki da się upakować wykorzystując nieograniczoną ilość każdej rzeczy R_i

1. Wyzeruj tablicę p[1..W]
2. Dla rzeczy o numerze i=1,2,..,n:
    dla pojemności j=1,2,..,W:
        jeśli rzecz o numerze i zmieści się w pojemności j to:
            //zdecyduj czy warto wziąć jedną rzecz nr i i dodać jej wartość
            //do optymalnego wypełnienia reszty przestrzeni plecaka
            //rzeczami rozpatrzonymi do tej pory
            P[j]=max(P[j], plecak[i-w_i]+ p_i)