**АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА  
РАЗЛИЧНИ СУМИ**

Числата от дадената редица записваме в a[0], a[1], …, a[n−1]. Най-малката възможна сума се получава като съберем всички отрицателни числа, а най-голямата – като съберем всички положителни числа от редицата. В програмата тези две стойности се пресмятат в m1 и m2. Понеже ще използваме абсолютната стойност на m1, заменяме m1 с –m1. Може да вземем m1=0, ако в дадената редица няма отрицателни числа и m2=0, ако няма положителни.

В масива T[] ще записваме T[j+m1]=1, само ако j може да се получи като сума на числа от подредица на дадената. Така работим с неотрицателни стойности на индекса в масива T[], защото всичките възможни суми j са диапазона от –m1 до m2.

Запълваме стойностите в масива T[], последователно разглеждайки за i = 0, 1, 2, …, n, възможните суми на подредици в редицата a[0], a[1], …, a[i].

Така за i=0 записваме T[a[0]+m1]=1 и след това за i > 0:

Ако a[i]<0, пробягваме j в растящ ред и записваме T[j+m1]=1, ако

T[j+m1-a[i]) е било равно на 1.

Ако a[i]>0, пробягваме j в намаляващ ред и записваме T[j+m1]=1, ако

T[j+m1-a[i]) е било равно на 1.

И в двата случая записваме T[a[i]+m1]=1.

Накрая преброяваме колко от елементите на масива T[] са равни на 1.

Описаният алгоритъм на динамичното оптимиране има сложност О(n(m1+m2)).

*Автор: Емил Келеведжиев*