Задача B2. МИНИМАЛНОСТ

Пояснение към решението

Програмата прочита числата от даденото множество в масива a[i], i=0, …, n−1 и натрупва сумата им в ts.

Решение с пълно изчерпване или с търсене с връщане може да работи при малки стойности на n, понеже броят на операциите расте експоненциално спрямо n.

Задачата в рамките на дадените ограничения може да бъде решена чрез динамично програмиране, понеже сумата ts на елементите от даденото множество не е много голяма и ако няма допълнително ограничение за памет е възможно да използваме булев масив t[N+1][TS+1], където N е максималният брой елементи в даденото множество, а TS е максималната сума от всички елементи. Ще запълваме t[i][j] така, че t[i][j]=true, точно когато някое подмножество, съставено от елементи измежду първите i е такова, че има сума равна на j. При последователното запълване по редове i=1,2, …, n, а във всеки ред по стълбове j=1, 2, …, ts, вземаме предвид, че за сума j елементът a[i−1] може да не участва и тогава t[i][j] = t[i − 1][j], или ако участва (при a[i−1] ≤ j), тогава гледаме дали t[i-1][j−a[i−1]] е true.

Накрая намираме най-голямата стойност на j, за която t[n][j]==true и отпечатваме ts−2*j.

При наличие на допълнително ограничение за памет трябва да пренапишем динамичното програмиране така, че да не се използва двумерен масив. Използваме два булеви едномерни масива d[T] и c[T], където T=ts/2+1. В процеса на пресмятането се използва спомагателен масив c[i], чиито true стойности се прехвърлят в d[i], за да може да бъде направена следваща итерация в основния цикъл for (int i = 0; i < n; i++). Стойността d[i] накрая ще покаже дали е възможно i да е сума от елементи на даденото множество. В основния цикъл някоя стойност c[k] става true за такъв индекс k, за който d[k-a[i]] е true.

Накрая, аналогично както при предната реализация, намираме най-голямата стойност на i, за която d[i] е true и отпечатваме ts−2*i.

Емил Келеведжиев