

Задача С?. Суми

Пояснение към решенията

Бавно решение.

Програмата прочита дадената редица в масива $a[]$, който е с дължина n . След това масивът се сортира и в m се пресмята сумата на елементите му. Използваме друг масив $t[]$ с дължина поне $m+1$. В j -тия елемент на този масив ще пресметнем по колко различни начина може да се образува сума със стойност j . Това става последователно по елементите на масива $a[]$. Първо записваме $t[a[1]]=1$ и след това използваме цикъла:

```
for(int i=2;i<=n;i++)
{
    for(int j=1;j<m;j++) tp[j]=t[j];
    for(int j=1;j<m;j++) t[j+a[i]] += tp[j];
    t[a[i]]++;
}
```

Броят на всички суми, които могат да се получат от елементите на входната редица се пресмята чрез:

```
int cc=0;
for(int j=0;j<=m;j++) if(t[j]>0){cc++;}
cout << cc << endl;
```

и този брой се извежда на първия ред от изхода.

Числата на втория ред от изхода се пресмятат и извеждат чрез

```
int v=0; long long k=0;
for(int j=0;j<=m;j++) if(k<=t[j]){k=t[j]; v=j;}
cout << v << " " << k << endl;
```

За да изведем третия ред от изхода използваме рекурсивната функция $go(v, 1)$, където v е разглежданата сума, за която търсим представяне чрез елементи от дадената редица, а вторият входен параметър, който първоначално е 1, се отнася за броя на елементите от дадената редица, които са използвани в процеса на търсене. В процеса на извикванията на функцията go всеки пореден елемент от дадената редица или се използва, или не се използва:

```

if (v-a[i] >= 0) {b[i]=a[i];go(v-a[i],i+1);b[i]=0;}
if(ex) return;
go(v,i+1);
if(ex) return;

```

Извикванията продължават когато не е достигнат края на дадената редица или когато не е намерена представяща сума:

```

if(v==0)
{
    bool f=false;
    for(int j=0;j<=n;j++)
        if(b[j]>0) {if(f) cout << " "; f=true; cout << b[j];}
    cout << endl;
    ex=true;
    return;
}

```

Когато се намери такава представяща сума, тя се отпечатва с използването на спомагателния масив `b[]`, след което чрез използване на булевата променлива `ex` се излиза от рекурсията. Начинът на построяването на рекурсията гарантира, че ще бъде намерена най-малката в лексикографски смисъл редица.

Това решение е бавно, защото чрез рекурсията се реализира търсене с връщане. Предимство е малката памет, която използва програмата, защото има само едномерни масиви.

Бързо решение

Сортираме числата от входа в намаляваща последователност `a[i]`.

Използваме двумерен масив `dp[MAXN][MAXS]`, в който броят на редовете е равен на броя на числата във входа и броят на стълбовете е равен на максималната сума, която може да се образува. Запълваме го последователно по редове. Искаме в `j`-тата позиция на `i`-ред стойността да показва по колко различни начина може да се образува сума равна на `j` чрез използване на първите `i` числа от входа. Така, ако вече сме запълнили редовете до `i-1`-вия включително, то за следващия ред пресмятаме:

```

int x = a[i];
for (int j = 0; j < MAXS; ++ j)

```

```

{
    dp[i][j] = dp[i - 1][j];
    if (j >= x) dp[i][j] += dp[i - 1][j - x];
}

```

В последния n-ти ред се получава информация, от която може да пресметнем стойностите, които да отпечатаме на първите два реда в изхода:

```

long long max_cnt = 0;
int s = 0, cnt = 0;
for (int j = 1; j < MAXS; ++ j)
{
    if (dp[n][j] != 0) cnt++;
    if (dp[n][j] >= max_cnt)
        { max_cnt = dp[n][j]; s = j;}
}
cout << cnt << endl;
cout << s << ' ' << max_cnt << endl;

```

За да намерим редицата, която представя сумата s и която трябва да изведем на третия ред в изхода, Обхождаме от долу нагоре по редове запълнения масив dp [MAXN] [MAXS]. Започваме с най-малката стойност x=a [i] и ако тази стойност е по-малка или равна на s, тогава x е първото число от търсената редица. Целият фрагмент, който намира търсената редица е следният:

```

for (int i = n; i >= 1; -- i)
{
    int x = a[i];
    if (s >= x and dp[i-1][s - x] > 0)
        {cout << x << ' '; s -= x;}
}

```

Начинът на построяването на обхождането гарантира, че ще бъде намерена най-малката в лексикографски смисъл редица с търсеното свойство.

Емил Келеведжиев и Добрин Башев