ДИнамично оптимиране

**Задача 2.**Написать программу определения количества шестизначных 'счастливых' билетов, у которых сумма первых 3 десятичных цифр равна сумме 3 последних десятичных цифр.

І начин: наивен, със сложност 106

**int Br=0;**

**for (int i=0; i<=9; i++)**

**for (int j=0; j<=9; j++)**

**for (int k=0; k<=9; k++)**

**for (int l=0; l<=9; l++)**

**for (int m=0; m<=9; m++)**

**for (int p=0; p<=9; p++)**

**if (i+j+k==l+m+p)Br++;**

**cout<< Br << endl;**

ІІ начин – намаляваме сложността на 105 като изразяваме последната цифра:

**int Br=0;**

**for (int i=0; i<=9; i++)**

**for (int j=0; j<=9; j++)**

**for (int k=0; k<=9; k++)**

**for (int l=0; l<=9; l++)**

**for (int m=0; m<=9; m++)**

**{**

**int p = (i+j+k)-(l+m);**

**if (p>=0 && p<=9)Br++;**

**}**

**cout<< Br << endl;**

ІІІ начин – намаляваме сложността на 103 като разглеждаме само първите 3 цифри и тяхната възможна сума. Означаваме я с Т, като Т е в [0, 27] /от 000 до 999/. Ще обходим от 0 до 27 и ще преборим колко билета ще имат такава сума, ще запазим резултата в масив. Ако а[i] е броя на билетите, със сума i на първите три цифри, то броя на щастливите билети ще е а[i]\*a[i]. Затова накрая обхождаме масива и събираме квадратите на елементите му.

int Br=0, a[28];

for (int i=0; i<28; i++) a[i]=0;

for (int i=0; i<=9; i++)

for (int j=0; j<=9; j++)

for (int k=0; k<=9; k++)

{

int T = i+j+k;

a[T]++;

}

for (int i=0; i<28; i++) Br=Br+a[i]\*a[i];

cout<< Br << endl;

ІV начин – можем да забележи, че колкото билета имат сума Т, то толкова ще имат и сума 27-Т, затова можем намерим сумата от квадратите на сумите от 0 до 13 и след това да удвоим резултата.

int Br=0, a[28];

for (int i=0; i<28; i++) a[i]=0;

for (int i=0; i<=9; i++)

for (int j=0; j<=9; j++)

for (int k=0; k<=9; k++)

{

int T = i+j+k;

a[T]++;

}

for (int i=0; i<14; i++) Br=Br+a[i]\*a[i];

cout<< 2\*Br << endl;

V начин:

В предходните два начина обхождаме и преброяваме комбинациите, които дават сума Т. В този начин ще ги пресметнем.

За всяка сума Т /ще разглеждаме от 0 до 13 и после ще удвоим резултата/ знаем, че T = i + j + k;

Пресмятаме в какъв интервал се променя i:

Минималната стойност на i се получава при j = 9 и k=9, затова минималната стойност е T – j – k, но тъй като не трябва да е по-малко от 0, затова вземаме 0, при отрицателно или стойността T – j – k, при положително, т.е. вземаме по-голямата от двете стойности. Това може да го изразим така:

минималната стойност на i е = max (0, T – j – k)

Максималната стойност се получава при j = 0 и k = 0, затова е равна на T, но не трябва да надвишава 9, затова вземаме по-малката от двете стойности, т.е.

максималната стойност на i е = min (9, T).

Аналогично, минималната стойност на j е = max (0, T – 9 – i), минималната стойност е по-малкото от 9 и T – i, т.е. min (9, T – i).

к се изразява еднозначно к = T – i – j.

Броят на комбинациите, започващи с i са толкова, колкото са стойностите на j, т.е. разлика на максималната и минималната стойност + 1, т.е max j – min j + 1, т.е. = min(9, T – i) + max (0, T – 9 – i) + 1. Обхождаме всички възможности за i от неговата минимална до неговата максимална стойност. Всички възможности за комбинация със сума Т запазваме в Р. След това добавяме Р\*Р към броя. Повтаряме всичко това за всички възможни суми Т от 0 до 13. Накрая удвояваме резултата.

*int Br=0;*

*for (int T = 0; T<=13; T++)*

*{*

*int P=0;*

*for (int i=max(0, T-18); i<=min(9, T); i++) P=P+min(9, T-i)-max(0, T-9-i)+1;*

*Br=Br+P\*P;*

*}*

*cout<< 2\*Br << endl;*