

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ВЕЗНИ

Математическата формулировка на задачата е следната:

Даден е масив от цели положителни числа  $a[0], a[1], \dots, a[n]$ .

Да се намери броят на тройките **различни** индекси  $i, j, k$ , за които  $a[i] + a[j] = a[k]$ .

При разместване на елементи в масива, отговорът не се променя, затова предварително сортираме числата в растящ ред и след това намираме броя на всички тройки индекси  $0 \leq i < j < k < n$ , за които е изпълнено  $a[i] + a[j] = a[k]$ .

От  $a[i] \leq a[j] \leq a[k]$  и  $a[i] + a[j] = a[k]$  следва, че  $2*a[i] \leq a[k]$  и  $2*a[0] \leq a[k]$ .

**Първи начин.** Използваме масив  $b[ ]$ , в който  $b[x]$  е равно на броя на сумите  $a[i] + a[j]$ , които са равни на  $x$ .

```
// Решение 1 -  $O(n^2)$ 
for(int i=0; 2*a[i]<=a[n-1]; i++)
    for(int j=i+1; a[i]+a[j]<=a[n-1]; j++)
        b[a[i]+a[j]]++;

long long br=0;
for(int k=2; k<n; k++)
    br = br + b[a[k]];
cout << br << endl;
```

**Втори начин.** Използваме два индекса, които тръгват от двата края на някакъв подмасив и се движат един срещу друг.

```
// Решение 2 -  $O(n^2)$ 
long long count(int x, int i, int j)
{ long long cnt = 0;
  while(i<j)
  { int s = a[i]+a[j];
    if(s<x) i++;
    if(s>x) j--;
    if(s==x)
    { if(a[i]<a[j])
      { int p=i+1; while(a[i]==a[p]) p++;
        int q=j-1; while(a[q]==a[j]) q--;
        cnt = cnt + 1LL*(p-i)*(j-q);
        i = p; j = q;
      }
      else
      { cnt = cnt + 1LL*(j-i+1)*(j-i)/2;
        return cnt;
      }
    }
  }
  return cnt;
}

long long br=0;
int k=lower_bound(a,a+n,2*a[0]) - a;
while(k<n)
{ long long s = count(a[k],0,k-1);
  int p=k+1;
  while(p<n and a[k]==a[p]) p++;
  br = br + (p-k)*s;
  k = p;
}
cout << br << endl;
```

**Трети начин.** Отначало може да се направи „свиване” на масива, като се пресметнат кратностите на всяко от числата (числото  $a[i]$  се среща  $c[i]$  пъти в оригиналния масив):

```
// Решение 3 -  $O(n^2)$ 
long long count(int x, int i, int j)
{ long long cnt = 0;
  while(i<j)
  { int s = a[i]+a[j];
    if(s<x) i++;
    if(s>x) j--;
    if(s==x)
    { cnt = cnt + 1LL*c[i]*c[j];
      i++; j--;
    }
  }
  if(2*a[i]==x) cnt = cnt + 1LL*c[i]*(c[i]-1)/2;
  return cnt;
}

// Свиване на масива
int m=0; // брой различни числа
int i=0;
while(i<n)
{ int j=i+1;
  while(j<n and a[i]==a[j]) j++;
  a[m]=a[i]; c[m]=j-i; m++;
  i = j;
}
n = m;

long long br=0;
int k0=lower_bound(a, a+n, 2*a[0]) - a;
for(int k=k0; k<n; k++)
  br = br + c[k]*count(a[k], 0, k-1);
cout << br << endl;
```

*Автор: Стоян Капралов*