Задача ИНДЕКС

Пояснение към решенията

Бавно решение – файл `index_bf_39p.cpp`

Програмата прочита в `a[0]`, `a[1]`, …, `a[n-1]` редицата от входния файл. Алгоритъмът използва brute force (груба сила): В цикъл с брояч `i`=0,1,…, n-1, се пресмятат стойностите `L` и `R` съгласно условието на задачата:

```
long long L = 0;
for (int j = 0; j < i; j++)
  L += a[j] * (i - j);

long long R = 0;
 for (int k = i + 1; k < n; k++)
   R += a[k] * (k - i);
```

и всеки път се проверява `if (L == R) {cout << i << endl; return 0;}`

Така програмата съдържа вложени цикли и броят на действията е от порядъка на n^2.

Бързо решение – файл `index.cpp`

За да направим решение за 100 точки, трябва да премахнем двойния цикъл в предното решение. За целта поддържаме 4 променливи `Ls`, `Rs`, `L` и `R`. Използва се един цикъл

```
for (int i = 1; i < n; i++)
```

В тялото на цикъла променливите `Ls`, `Rs`, `L` и `R` се актуализират така, че:

`Ls` е сумата `a[0]+…+a[i-1]`. В началото `Ls=0`.

`Rs` е сумата `a[i+1]+…+a[n-1]`. В началото `Rs` е сумата `a[1]+…+a[n-1]`.

`Ls` на всяка стъпка се актуализира чрез `Ls += a[i-1]`.

`Rs` на всяка стъпка се актуализира чрез `Rs -= a[i]`.

`L` при всяка стъпка на цикъла има същата стойност, каквато е стойността на `L` от съответната стъпка на външния цикъл при програмата `index_bf_39p.cpp`. Сега това се извършва чрез оператора `L += Ls;`

`R` при всяка стъпка на цикъла има същата стойност, каквато е стойността на `R` от съответната стъпка на външния цикъл при програмата `index_bf_39p.cpp`. Сега това се извършва чрез оператора `R -= Rs;`

Важен е редът в който се актуализират променливите `Ls`, `Rs`, `L` и `R`: Първо се актуализира `Ls` и след това последователно `L` и `R`, и накрая `Rs`.

И както при програмата `index_bf_39p.cpp` на всяка стъпка се проверява

```
if (L == R) {cout << i << endl; return 0;}
```

Зорница Дженкова