

Задача ТРОЙКИ

Пояснение към решенията

Бавно решение – файл triplet_18p.cpp

Програмата прочита във вектора $s[]$ елементите на дадената редица. Вместо вектор би могло да се използва и масив. След това елементите на редицата се сортират в растящ ред. Намирането на питагорова тройка става чрез троен цикъл с индекси i, j и k . В тялото на тройния цикъл се проверява дали квадратите на числата $a=s[i], b=s[j], c=s[k]$ образуват питагорова тройка, т.е. дали $a^2+b^2=c^2$. Първата така намерена тройка удовлетворява условието на задачата.

По-бързо решение – файл triplet_68p.cpp

След прочитане на дадената редица в $s[]$, заменяме всяка стойност с нейния квадрат. Сортираме елементите на $s[]$ и използваме цикъл с индекс i , с който обхождаме елементите, започвайки от $i=2$. При всяка стойност на i обработваме отреза от $s[]$ с индекси от L до R . В началото $L=0$ и $R=i-1$.

Когато $s[L] + s[R] == s[i]$, това означава, че е намерена питагорова тройка и тогава в променливите ar, br и cr записваме квадратите на числата от питагоровата тройка, като се поддържа ar да е най-малката намерена до момента стойност, а когато има повече от една питагорова тройка с такава най-малка стойност, в променливата br се поддържа най-малката намерена стойност на b .

Когато $s[L] + s[R]$ не е равно на $s[i]$, тогава при $s[L] + s[R] < s[i]$ увеличаваме L с единица, а в противен случай – намаляваме R с единица. Така се поддържа, че $L < R$ и $R < i$, докато е възможно.

Накрая се отпечатва

```
cout << sqrt(ar) << " " << sqrt(br) << " " << sqrt(cr) << endl;
```

или, когато не е намерена питагорова тройка, `cout << -1 << endl;`

Забележка. С малка модификация може да не използваме математическата функция `sqrt`. След прочитане на дадената редица в $u[]$, записваме за всяка нейна стойност нейния квадрат в $s[]$. Така вместо да отпечатваме

```
cout << sqrt(ar) << " " << sqrt(br) << " " << sqrt(cr) << endl;
```

отпечатваме

```
cout << u[ai] << " " << u[bi] << " " << u[ci] << endl;
```

което е направено във файла `triplet_68p_alt.cpp`

Бързо решение – файл triplet_100p.cpp

Означаваме с m най-голямата стойност на елемент от дадената редица. Не използваме сортиране, а масив (или вектор) $vis[]$, в който отбелязваме $vis[a]=true$, когато стойността a се среща в дадената редица и $vis[a]=false$, когато не се среща. В тялото на цикъла

```
for(LL a = 1; a < m + 1; a++)
```

ще изследваме, дали всяко от числата a в интервала от 1 до m може да бъде първи елемент на питагорова тройка (съгласно ограниченията на задачата, може вместо m да използваме стойността 7 000). За целта във вътрешния цикъл `for(LL b = a+1; b < m+1; b++)` проверяваме дали $a^2 + b^2$ е точен квадрат. Това става чрез математическата функция `sqrt`: `c = sqrt(a*a + b*b)`. Проверяваме дали `c*c==a*a+b*b` и дали c е елемент на дадената редица. Ако е така, стойностите a, b, c са търсената питагорова тройка.

Емил Келеведжиев