

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА НАЙ-МАЛКО ЧИСЛО

Тъй като на този етап от обучението си, състезателите от група Е не са задължени да владеят работа с масиви, първо давам две решения без използването на масиви. В края на анализа съм дал и решение с използването на масив, което може да се разгледа след като учениците усвоят съответните познания.

Първо решение без масиви

Първото нещо, което ще направим, е да подредим трите числа A , B и C в ненамаляващ ред, т.е. да сме сигурни, че $A \leq B \leq C$. Тогава можем да направим следния алгоритъм:

1. Ако $A > 1$, то 1 е най-малкото число, което не може да се представи като сума на някои от трите числа. **Край на алгоритъма.**
2. Продължава се при условие $A = 1$.
3. Ако $B > 2$, то 2 е най-малкото число, което не може да се представи като сума на някои от трите числа. **Край на алгоритъма.**
4. Продължава се при условия $A = 1$, $B \leq 2$.
5. Условие $A = 1$, $B = 1$. Ако $C > 3$, то 3 е най-малкото число, което не може да се представи като сума на някои от трите числа. **Край на алгоритъма.**
6. Продължава се при условия $A = 1$, $B = 1$, $C \leq 3$.
7. Условие $A = 1$, $B = 1$, $C = 1$. Най-малкото число, което не може да се представи като сума на някои от трите числа е 4. **Край на алгоритъма**
8. Условие $A = 1$, $B = 1$, $C = 2$. Най-малкото число, което не може да се представи като сума на някои от трите числа е 5. **Край на алгоритъма**
9. Условие $A = 1$, $B = 1$, $C = 3$. Най-малкото число, което не може да се представи като сума на някои от трите числа е 6. **Край на алгоритъма**
10. Условие $A = 1$, $B = 2$. Ако $C > 4$, то 4 е най-малкото число, което не може да се представи като сума на някои от трите числа. **Край на алгоритъма**
11. Продължава се при условия $A = 1$, $B = 2$, $C \leq 4$.
12. Условие $A = 1$, $B = 2$, $C = 2$. Най-малкото число, което не може да се представи като сума на някои от трите числа е 6. **Край на алгоритъма**
13. Условие $A = 1$, $B = 2$, $C = 3$. Най-малкото число, което не може да се представи като сума на някои от трите числа е 7. **Край на алгоритъма**
14. Условие $A = 1$, $B = 2$, $C = 4$. Най-малкото число, което не може да се представи като сума на някои от трите числа е 8. **Край на алгоритъма**

Ако се вгледате по-внимателно в отговорите, ще видите, че в случаите 7,8,9,12,13 и 14 отговорът е просто $A+B+C+1$.

Както се вижда, възможните отговори са числата от 1 до 8, което увеличава опасността от случайно „познаване“ на отговора. Това е причината всеки тест да съдържа три примера. Най-удачно е да се направи цикъл от 1 до 3, като в тялото на цикъла се четат трите числа от поредния ред на входа, изпълнява се горния алгоритъм и се извежда на отделен ред резултатът.

Това решение е реализирано във файл **minnum1.cpp**.

Второ решение без масиви

Не се предполага подредба на трите числа. В четири допълнителни променливи, напр. $S1$, $S2$, $S3$ и $S4$ се образуват сумите

$S1=A+B;$
 $S2=B+C;$
 $S3=A+C;$
 $S4=A+B+C;$

Тези четири променливи заедно с променливите A , B и C съдържат всички възможни числа, които могат да се получат като суми на някои от числата A , B и C .

След това се пуска цикъл, в който някаква променлива, напр. i , приема стойности от 1 до $A+B+C+1$. За всяка стойност на i се проверява дали тя не е равна на стойността на някоя от променливите A , B , C , $S1$, $S2$, $S3$, $S4$. Първата стойност на i , която не е равна на стойността на нито една от променливите, е търсеното минимално число.

Внимание: При намирането на минималното i , което не е равно на нито една от променливите A , B , C , $S1$, $S2$, $S3$, $S4$ (т.е. при намирането на отговора на задачата), цикълът трябва да се прекъсне, тъй като сумата $A+B+C+1$ може да бъде много голяма (вижте ограниченията в условието на задачата) и да се стигне до превишаване на ограничението по време. Вижте предното решение и ще разберете, че най-много за 8 завъртания на цикъла решението ще бъде намерено.

Това решение е реализирано във файл **minnum2.cpp**.

Решение с масив

Решението с масив напомня много второто решение без масив, само че в този случай седемте възможни суми на някои от числата A , B и C се записват в масив, а не в единични променливи. Тогава проверката дали поредното i е равно на някоя от сумите става по-лесно (или поне с по-малко *if*-ове) като се направи вложен цикъл.

Автор: Руско Шиков