Анализ на задача `alley`

Тагове: аритметични прогресии, наблюдения, STL, умно обхождане

По същество, в тази задача е дадена редица от числа и трябва да проверим дали може да я разделим на две непресичащи се аритметични прогресии.

Решение на втора подзадача - 20 точки

Тук няма нищо за измисляне единствено за имплементация, но въпреки това има участници, които не са изкарали тези точки, а е хубаво да се вземат такива лесни точки. Ограничението за N е достатъчно малко, така че можем да използваме пълно изчерпване по подмножества и да разгледаме всички възможност за това кои числа от началната редица са в I аритметична прогресия. Останалите числа трябва да са в другата аритметична прогресия и остава само за линейно време да проверим дали двете редици са наистина аритметични прогресии. Ако сортираме предварително числата, то няма нужда да правим каквото и да е сортиране на числата в редиците при пълното изчерпване, защото можем лесно да ги обхождаме в сортирания ред.

Сложност: $O(N \log_2 N + 2^N N)$.

Решение на трета подзадача - 50 точки

От сега нататък ще мислим, че в началото сме сортирали всички числа в редицата. Ясно е, че една от двете аритметични прогресии ще включва минималното число на редицата като първо. БОО (Без ограничения на общността) нека това е I редица. Като сме фиксирали първото число в I прогресия, можем да разгледаме всички възможности за кое число ще е второ в тази редица. По този начин ще сме фиксирали и разликата на тази аритметична прогресия. Така лесно вече можем да преценим дали число с по-голяма стойност е възможно да се добави към тази прогресия, като проверим дали разлика му с последното добавено е равна на разликата на прогресията.

Имаме следното лесно наблюдение. Нека разглеждаме останалите числа (с не по-малка стойност от първите две фиксирани на прогресията) на редицата едно по едно. Тогава можем да добавяме числата на точната разлика от последното включено в аритметичната прогресия, но ако някъде не добавим число с такава подходяща стойност, то вече няма да можем да включваме числа с по-големи стойности (разликата между тях и последното включено в аритметичната прогресия ще стане много голяма). Понеже ограничението за N в тази подзадача е по-малко, то това е достатъчно да реализираме по-добра идея - фиксираме кое число ще е второ в I аритметична прогресия, след което добавяме едно по едно подходящи числа към прогресията и всеки път проверяваме дали останалите числа (във втората редица) образуват аритметична прогресия с линейно обхождане.

Сложност: $O(N \log_2 N + N^3)$.

Решение на четвърта подзадача - 20 точки

В тази подзадача можем допълнително да намалим проверките за отговор (досега те бяха около N^2). Нека сортираните числа са $a[0], a[1], a[2], \ldots$. Аналогично на разсъждението за минималното число в редицата ($a[0]$), можем да погледнем и $a[1]$. За него имаме две възможности - или е второто число на I аритметична прогресия, или е първото числото на II аритметична прогресия. Ако случайно е първото число на II аритметична прогерсия, то тогава $a[2]$ ще е второ число или в I, или във II аритметична прогресия. Нека разгледаме какви възможности получаваме за първи две числа за някоя аритметична прогресия. В единия случай, I аритметична прогресия има за първи две числа $(a[0], a[1])$ или $(a[0], a[2])$, а в другия случай II аритметична прогресия има за първи две числа $(a[1], a[2])$. Така можем да направим следното наблюдение -

една от аритметичните прогресия има за първи две числа някоя двойка числа от $a[0], a[1], a[2]$.

Това означава, че сега за фиксираната I аритметична прогресия можем да разгледаме само три възможности за първите две числа (вместо $N-1$, както беше преди). Като използваме помощното ограничение на подзадачата, че има решение, където аритметичните прогресии са с равна дължина (или казано по друг начин - дължината им е $N/2$), то вместо да разглеждаме всички случаи за това до къде да добавим още числа към тази аритметична прогресия, то можем да разгледаме само случая, в който тя ще стане с големина $N/2$ (ако въобще има такъв). Така всички случаи за I аритметична прогресия стават най-много 3 на брой и можем линейно да проверим дали другата редица ще излезе аритметична прогресия Да обърнем внимание и на частния случай $N = 2$, когато винаги има решение, но нямаме $a[2]$ и е добре да го обработим отделно в началото.

Сложност: $O(N \log_2 N)$ (сложността се доминира от сортирането в началото).

Решение на пета подзадача - 100 точки

Единственото, което не сме оптимизирали достатъчно е проверката за това дали числата за II редица образуват аритметична прогресия. Достатъчно е да знаем дали разликата между всеки две съседни по големина числа са еднакви. Най-лесно това можем да правим като поддържаме един *multiset* с разликите и гледаме дали първото и последното му число съвпадат. Можем да си мислим, че в началото всички нефиксирани числа от началната редица са част от втората редица и малко по-малко прехвърляме от тях към първата. Всеки път като добавим ново число към I аритметична прогресия трябва да изтрием разликите му със съседните по големина числа от II редица (понеже сме сортирали числата от начало, то лесно можем да знаем кои са те) и евентуално да добавим една нова разлика между двете съседни числа на изтритото, които вече ще станат едно до друго. Така ще имаме сложност $O(N \log_2 N)$, за да проверим една възможност за решение. Ограничението по време е достатъчно голямо, така че и по-тежки реализации да минават задачата, а и в началото така или иначе трябва да сортираме началната редица.

Алтернативна идея е да използваме подхода умно обхождане, за да знаем лесно дали имаме една единствена разлика и за линейно време да можем да проверим една възможност за решение. Но няма смисъл да използваме този малко по-пипкав подход при дадените ограничения на задачата, затова няма да посочваме детайли.

Сложност: $O(N \log_2 N)$.

Автор: Илиян Йорданов (заета)