Arithmetic Subsequence

Problem G

Chin Ka Wang {rina__owo}

The 1st Buddhist Sin Tak College Computer Club Programming Contest

October 18, 2024

stpc();

Background

Problem Idea by `rina__owo`
Preparation by `rina__owo`, `pepper1208`

Background Information

In mathematics, an **arithmetic sequence** is a special sequence where the difference between any two consecutive terms is constant. This difference is called the common difference. For example, the sequence $\{1, 3, 5\}$ is an arithmetic sequence with a common difference of 2, and the sequence $\{1, -4, -9, -14\}$ is an arithmetic sequence with a common difference of -5. The common difference can be any real number.

Problem Restatement

Given an array containing N integers.

if a **continuous** k numbers within the array form an **arithmetic sequence** without changing their order, we call it an **arithmetic subsequence**.

Output the largest k among all **arithmetic subsequences** of the input array.

Statistics

- Points are given per subtask in this problem. There are 3 subtasks in this question.

  Attempts: 43

0 points	12	+	0	=	12
Subtask 1 (21 points)	0	+	3	=	3
Subtask 2 (28 points)	0	+	3	=	3
Subtask 3 (51 points)	0	+	2	=	2

  First solved by **Chan Tsz Hang** at **31m 34s**

Subtasks

Subtask	Score	$N \leq$	n_i	$k \leq$
1	21	10	$n_i > 0$	3
2	28	100	$n_i > 0$	100
3	51	1000	$-1000 \leq n_i \leq 1000$	1000

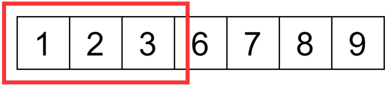
Solution

Observe that an arithmetic sequence is formed by consecutive numbers, the given array must be able to be divided into different parts with common head and tail, where each part is a arithmetic sequence.

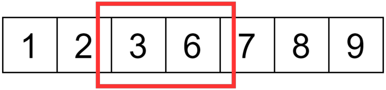
1	2	3	6	7	8	9

Sample 1

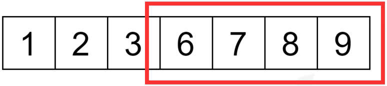
Solution



The first part



The second part



The third part

Solution

- As the difference between any two consecutive terms in a arithmetic sequence is constant, we can set up a variable `cd` to store the current common difference between the current consecutive terms.

  Once the common difference does not match the previous one, we know that the arithmetic sequence has ended and a new arithmetic sequence is formed as a new common difference appeared. The value of `cd` should then be updated to be the new common difference.

Solution

Note that the detection is only related to the current consecutive numbers, so we can just store the numbers by two variables `prev` and `curr` instead of an array.

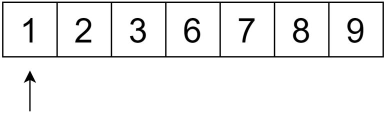
Set up a variable `ans` to store the largest *k*, as known as the length of the longest arithmetic subsequence.

Each time we detect a number belongs to the arithmetic subsequence, increase the *k* of the required subsequence by one.

Keep updating `ans` by comparing the current *k* and current `ans`.

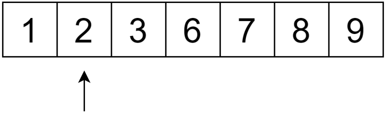
Reset the value of *k* to be 2 after an arithmetic subsequence ended.

Simulation



1	2	3	6	7	8	9

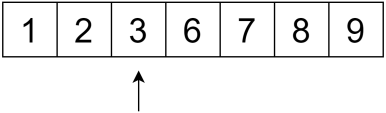
$$k = 1,\ \texttt{cd} = 0,\ \texttt{ans} = 1$$



1	2	3	6	7	8	9

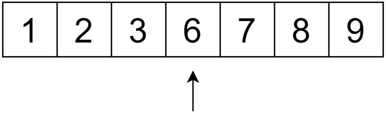
$$k = 2,\ \texttt{cd} = 1,\ \texttt{ans} = 2,$$

Simulation



1	2	3	6	7	8	9

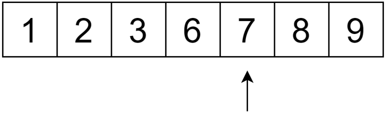
$k = 3$, $\texttt{cd} = 1$, $\texttt{ans} = 3$



1	2	3	6	7	8	9

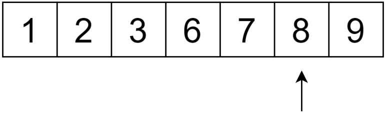
$k = 2$, $\texttt{cd} = 3$, $\texttt{ans} = 3$

Simulation



1	2	3	6	7	8	9

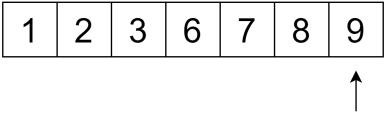
$$k = 2,\ \texttt{cd} = 1,\ \texttt{ans} = 3$$



1	2	3	6	7	8	9

$$k = 3,\ \texttt{cd} = 1,\ \texttt{ans} = 3$$

Simulation



1	2	3	6	7	8	9

$k = 4$, $\texttt{cd} = 1$, $\texttt{ans} = 4$

Output: 4

Takeaways

1. Be careful of the reset of the variables.
2. Better don’t waste the memory by creating an array if it is unnecessary.