Upon generating a final dataset, these were issues I had to resolve. I am totally unsure impact it will have elsewhere...

TICKER: A(2)A(2)-D(2)SSD(2)SS										
[35.0->35.1, 35.1->35.2, 35.2->35.1, 35.1, 35.2->35.2, 35.2->35.1, 35.1, 35.1]										
35.0f	, 35.1f, 3	5.1f, 35.2	f, 35.1f	, 35.1f,	/*	*/35.1f,	35.2f, 3	5.1f, 35	.1f, 35.1f	
	This	section is fine								
IJ	IK	IL	IM	IN	10	IP	IQ	IR	15	IT IU
35.1f	35.1f	35.2f	35.1f	35.1f	35.1f	35.2f	35.1f	35.1f	35.1f 35.	2f 21.1f
A	A	D	D	5	^ /	D	D	° /		3
1	35.1->35.2	35.2->35.1		35.1	35.2->35	.2 35.2->3	5.1	35.:	1 35.1->35.2	21.1
	CHI Inn This cou COU Pre- 2TT 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 199 197 197	ECKING: 35.1 wit ext number not d is is counter at unter is not zer UNTER VALUE: 1 evious number AN ECKING: 35.2 wit scending sequence ransition number 201982Using stor 7618cWriting ran exablishing start ECKING: 35.1 wit PEAT NUMBER 010Writing range ECKING: 35.1 wit PEAT NUMBER 010Writing 51.1 wit PEAT NUMBER 774544Writing 51 9238475Writing 5	h 35.2 escending (d: the moment: o D/OR next num h 35.1 e (difference : 35.2 descer ded start ge: 35.2 cescer ed start ge: 35.2 -> 35. h 35.1 23229USING : 35.2-> 35.1 h 35.1 andalone: 35 tandalone: 35	<pre>ifference)     1  ifference)     1  wber not ascence e)12Store nding(difference) .2 2)A(2)-D(2)SSD STORED TO WRIT 1 2)A(2)-D(2)SSD( .1 2)A(2)-D(2)SSD( 5.1 2)A(2)-D(2)-D(2)SD( 5.1 2)A(2)-D(2)-D(2)SSD( 5.1 2)A(2)-D(2)-D(2)SSD( 5.1 2)A(2)-D(2)-D(2)SSD( 5.1 2)A(2)-D(2)-D(2)-D(2)-D(2)-D(2)-D(2)-D(2)-D</pre>	ding(differen ed start -> e ce) on either red start -> TE RANGE (2) (2)5 (2)55	This is co ce) nd: 35.2->35. side end: 35.2->35	we reas	ok e can see for a on, it has not store	an unknown written to the	Jid have

## I visited this whole section of code again



TICKER: A(2)A(2)-D(2)SA(2)-D(2)SS [35.0->35.1, 35.1->35.2, 35.2->35.1, 35.1, 35.1->35.2, 35.2->35.1, 35.1] 35.0f, 35.1f, 35.1f, 35.2f, 35.1f, 35.1f, /\* ---\*/35.1f, 35.2f, 35.1f, 35.1f, 35.1f

ALL ISSUES ARE NOW FIXED

		KW	KX	KY	KZ	LA	LB	LC	LD	
		9.5f	9.4f	9.4f	9.5f	9.6f	9.6f	9.6f	9.7f	
		D	D	А	Α	А	Α	Α	Α	
	0.9	9.5->9.4		MISSING	9.5->9.6					
0.9	0.9f, 0.9f, 0.9f, 0.9f, 9.5f, 9.4f, 9.4f, 9.5f, 9.6f, 9.6f, 9.6f, 9.7f, 60.1f									
T	TICKER: SSSSD(2)A(2)SA(2)S									
[(	[0.9, 0.9, 0.9, 0.9, 9.5->9.4, 9.4->9.6, 9.6, 9.6->9.7, 60.1]									
It seems this issue is now resolved also due to above fixes										

I will now focus on the next area in the ChatGPT where there seems to be an error

Unfortunately I will need to run the entire 3000 numbers again.

I found a new issue and it all related to the new code I implemented, it needed a bit stringent conditions...

	JZ	KA	KB	KC	KD	KE	KF	KG	
	04 54	C1 44	C1 44	C1 45	00.05	50.04	50.04	50.04	
	61.5f	61.4f D	61.4f S	61.4f S	22.0f	59.6f S	59.6f S	59.6f	
	5 61.5->61.	4	61.4	61.4	22	59.6	59.6	59.6	
	61.5f D	61.4f D	61.4f S	61.4f S	22.0f S	59.6f S	59.6f S	59.6f S	
Í	5 61.5->61.	4	61.4	61.4->61.4	22				
			· · · · ·	61.4					
	21	21	01 04010	10 0	10 0	10 0	10 0	A1 A	
	1.1f, 1.	.1f, 1.2f	, 61.5f,	61.5f, 61	.5f, 61.4	4f, 61.4f	f, 61.4f,	22.0f, 59.6f, 59.6 <sup>4</sup>	f
	TTCKER	54(2)550	(2)555555						
	[1.1, 1	.1->1.2,	61.5, 61.	5, 61.5->6	51.4, <u>61.</u>	4, 61.4->	61.4, 61	4, 22.0, 59.6, 59.6	]
		CHECKING:	61.5 with 61.	5	This	is ok			
		REPEAT NU 019238475	MBER Writing Standa	lone: 61.5	THIS	IS UK			
		*******	**TICKER*****	***: SA(2)SS					
		CHECKING: Descendin	61.5 with 61. g sequence (di	4 fference)		IS OK	N61 4		
		Establish	ing start: 61.	5		/ enu. 01.5	-701.4		
This is ok		CHECKING: REPEAT NU	61.4 with 61. MBER	4	This is c	ok			
if (k>0)		-1010Writ	232 ing range: 61.	29USING STORED	TO WRITE DAM				
if (nums[k]==nums[k-1]    nums[k]==nums[k+1]) {		1	**TICKER*****	***: SA(2)SSD(2	)	de	From here	in, we should be tandalone writes of	
<pre>sm.add(start); System.out.println("019238475Writing Standalone;</pre>	: " + start);	CHECKING: REPEAT NU	61.4 with 61. MBER	4			But there ar	61.4	
<pre>completellcker(start, start,k,lengthNums); hasWrittenRepeatNumber=true;</pre>		019238475	Writing Standa **TICKER******	lone: 61.4 ***: SA(2)SSD(2	)5	writ	tes and als	o 61.4->6.14 which	
	2.0	CHECKING:	61.4 with 22.	0			15 1	nconect	
is larger difference, so it writes the range. I think I should only write range hasWrittenRepeatNumber = false	ge if	This is counter i	per not descen ounter at the s not zero	aing (aitterence moment: 1	=)				
Since otherwise it would write incorrect entry as to the right whe has captured start=nums[k-1] and end = nums[k] where both has a start = nums[k] where both has a s	ere it ve	COUNTER V previous	ALUE: 1 number AND/OR	next number not	ascending(dif	fference)			
same values		2bWriting	range: 61.4->	61.4	Lo	aic is flawe	ed here		
<pre>if (!(Math.abs(nums[k] - (nums[k+1] - difference)) <epsilon) !(math.abs(nums[k]="" &&="" (nums[k+1]="" +="" -="" <epsilon))<="" difference))="" pre=""></epsilon)></pre>		8888888888	Writing Standa	<pre>&gt;&gt;++: SA(2)SSD(2) alone: 61.4 ***: SA(2)SSD(2)</pre>	555	5			
{     System.out.println(potentialfurtherAscendingBeyondThisSt	art);	CHECKING:	22.0 with 59.6	5					
System.out.println(potentialfurtherAscendingBeyondThisEn System.out.println("2bWriting range: " + start + "-> " +	nd); end);	1next numb This is co	per not descend punter at the m	ding (difference noment: 0					
<pre>sm.add(start+"-&gt;"+end); completeTicker(start,end,k,lengthNums);</pre>		2next numb	er not descend number ascendir	ling (difference ng (difference)	:) OR next numbe	r descending(d	difference)		
}		*********	*TICKER*****	***: SA(2)SSD(2)	SSSS				
This is also part of the code that I adjusted		CHECKING: REPEAT NUM	59.6 with 59.6 IBER						
run untouched given above modification		07774544Wr	TICKER******	one: 59.6 ***: SA(2)SSD(2)	SSSSS				
<pre>//I no longer feel this is an else associated with below if //alse</pre>	,	019238475	TICKER	lone: 59.6	SSSSSS				
//{			**TICKER******	***: SA(2)SSD(2)	SSSSSS				
<pre>&amp;&amp; !(Math.abs(nums[k] - (nums[k+1] - difference)) <epsilo !(math.abs(nums[k]="" &&="" (nums[k+1]="" +="" -="" <epsilon<="" difference))="" pre=""></epsilo></pre>	n))	[1.1, 1.1- Standalone	>1.2, 61.5, 61 numbers: 9	.5, 61.5->61.4, Ascending chain	61.4, 61.4->	61.4, 61.4, 22 ding chains: 1	2.0, 59.6, 59. L TOTAL:	5]	
<pre>{    start = String.valueOf(nums[k]);</pre>		Transition	events: 0						
<pre>sm.add(start); System.out.println("8888888888Writing Standalone: " +</pre>	start);								
<pre>completeTicker(start, start,k,lengthNums); }</pre>									

## [1.1, 1.1->1.2, 61.5, 61.5, 61.5->61.4, 61.4, 61.4, 22.0, 59.6, 59.6]

It appears that it has completed the ChatGPT extract 3 successfully. However I was extremely aware that dataset did not include two consecutive transition elements such as Ascending, Descending, Ascending

So I am going through all my test cases and found an issue as follows



I am still quite doubtful on all my test cases when it reaches k==nums.length-2

Once I have finished test cases, I will devise a few more in which it has to make various decisions at this point based on previous numbers...

## At the moment, I have found a failed test case as below:

CHECKING: 4.7 with 4.8	As I examined this code, I could see I set the value hasWrittenRepeatNumber=true							
1next number not descending (difference)								
This is counter at the moment: 0	But this area of code is right at the bottom And it has not been set back to false. From understanding of my code, I set these booleans to enforce mutual exclusiveness during a single iteration of the k loop. So I think commenting this out is a good decision.							
next number ascending (difference)								
OStored start -> end: 4.7->4.8								
CHECKING: 4.8 with 4.9	1497	if((Math.abs(nums[k] - (nums[k-1] - difference)) <epsilon)< th=""></epsilon)<>						
<pre>Inext number not descending (difference)</pre>	1498	<pre>[[ (Math.abs(nums[k] - (nums[k-1] + difference)) <epsilon))< pre=""></epsilon))<></pre>						
This is counter at the moment: 0	1499 -							
2next number not descending (difference)	1500	<pre>if ((Math.abs(nums[k] - (nums[k-1] + difference)) <epsilon))< pre=""></epsilon))<></pre>						
previous number descending(difference) AND/OR next number ascending (difference)	1501 -							
Next number ascending	1502	ascendingCounter++						
Previous number descending (difference) AND next number ascending (difference)								
CHECKING: 4.9 with 5.0		cm add/natartialfunthanAccondingDourandThicStart," \", nums[[1]).						
<pre>1next number not descending (difference)</pre>		Sm. adu(potentialTurtuerAstendingSeyOnd HitSstart+ -> +hums[K]);						
This is counter at the moment: 0		System.out.printin(						
2next number not descending (difference)	1507	System.out.println("-1010Writing range: " + potential+urtherAscendingBeyondThis						
<pre>previous number descending(difference) AND/OR next number ascending (difference)</pre>		completeTicker(potentialfurtherAscendingBeyondThisStart,String.valueOf(nums[k]),						
Next number ascending		//hasWrittenRepeatNumber=true; //late change						
Previous number descending (difference) AND next number ascending (difference)								
		<pre>potentialfurtherAscendingBeyondThisStart="";</pre>						
CHECKING: 5.0 with 5.0		<pre>potentialfurtherAscendingBeyondThisEnd="";</pre>						
REPEAT NUMBER								
23229USING STORED TO WRITE RANGE	1514	<pre>if (String.valueOf(nums[k+1]).equals(potentialfurtherAscendingBeyondThisEnd))</pre>						
-1010Writing range: 4.7-> 5.0	1515	{						
	1516							
CHECKING: 5.0 with 4.9								
Descending sequence (difference)		$TICKER: \ A(4)D(2)$						
		[4, 7-5, 0, 5, 0-54, 9]						

I consider this to be a highly critical change and I will run through test cases again from the top..



I am finding that all issues occurring are related to the new code I introduced in this document. But I seem to be resolving them...

Unfortunately I discovered one of my changes above was detrimental, but I found a solid workaround... It is now a case of undoing a change (see overleaf).

	I found these issues when testing ChatG Once again it appears as fundamental is	GPT extract 3 again sue, but I will try	65.0f A	65.1f A	65.1f A	65.2f 9 A S	1.4f 9: S	1.4f	
	investigate.								
CHECKING: 93.7 with 65.0 Inext number not descending (diff This is counter at the moment: 0	erence)		65.0->65.1	1	65.1->65.2		91.4	91.4	
2next number NOT within differenc 5Writing standalone: 93.7	e OK		65.0->65.1	1	65.1->65.2		91.4	91.4	
CHECKING: 65.0 with 65.1 Inext number not descending (diff	erence)								
This is counter at the moment: 0 2next number not descending (diff	erence)		65.0->65.	65.1	65.1->65.2		91.4	91.4	
previous number descending(differ Next number ascending	ence) AND/OR next number ascending (diffe	rence)	35.0f	35.1f	35.1f	35.2f	35.1f	35.1f	
	22Stored start -> end: 65.0->65.1		~	A	A	A		3	
CHECKING: 65.1 with 65.1 REPEAT NUMBER			35.0->35.	1	35.1->35.2	2 35.2->35.	1		
23229USING ST -1010Writing range: 65.0-> 65.1	ORED TO WRITE RANGE								
CHECKING: 65.1 with 65.2 1next number not descending (diff	erence)		35.0->35.	1	35.1->35.2	2 35.2->35.	1		
This is counter at the moment: 0 2next number not descending (diff	erence)		35.0->35.	35.3	35.1->35.1	2 35.2->35.	1		
previous number descending(differ Previous number same as current	ence) AND/OR next number ascending (diffe	rence)			•				
6Writing Standalone: 65.1	boolean previously was wrong c	hoice 93.	7f, 65.0f,	, 65.1f,	65.1f,	65.2f, 9	1.4f, 91	.4f	
CHECKING: 65 2 with 91 4	22Stored start -> end: 65.1->65.2	[93.	7,65.0->	65.1 <b>,</b> 6	5.1, 65.	1->65.2,	91.4, 9	91.4]	
Inext number not descending (diff	Ference)								
2next number not descending (diff	Ference)	Sm.	add(potenti	ialfurth	erAscendi	ngBeyond	ThisStart		
previous number descending(differ next item is NOT ascending(differ	rence) AND/OR next number ascending (diffe rence)	rence) Sys	tem.out.pri	intln("-	1010Writi	ng range	: " + pot		
using stored start		com	pleteTicker	r(potent	ialfurthe	rAscendi	ngBeyondT	K=nu	ums.length-2
		//h	asWrittenRe	epeatNum	ber=true;				
REPEAT NUMBER		lf I e	nable this, then the	ere are no iss	ues Just to rec	ap why I			
019238475Writing Standalone: 91.4	1								
TICKER: SA(2)SA(2)SS						- 1	4./+,4	.81,4.91,5.01,	5.0 <del>1</del> ,4.9 <del>1</del>
Standalone numbers: 4 Ascending Transition events: 0	g chains: 2 Descending chains: 0 TO	TAL: 6				- 1	simply	write	
						_		[4.7	-73.0]
CHECKING: 4.7 with 4.8 Inext number not descending (d This is counter at the moment:	lifference) Ø	As I examined this code, I But this area of code is rigi	could see I set the	e value hasW	rittenRepeatNu been set back t	o false.	unio e e in ele la	and an address to	So I think I need to
next number ascending (differe	ence) 0Stored start -> end: 4.7->4.8	loop. So I think commenting this	out is a good deci	ision.	enforce mutual e	axciusiveness a	uring a single li	eration of the K	resolve situation in the area of code
CHECKING: 4.8 with 4.9 Inext number not descending (d	lifference)	1497 1498	if((Math.abs    (Math.abs	(nums[k] - ( (nums[k] - (	nums[k-1] - di nums[k-1] + di	fference)) <ep fference)) <ep< td=""><td>silon) silon))</td><td></td><td>for k=nums.length- 2 and create a</td></ep<></ep 	silon) silon))		for k=nums.length- 2 and create a
This is counter at the moment: 2next number not descending (d previous number descending(dif	0 lifference) Ference) AND/OR next number ascending (difference)		{ if ((Mat	h.abs(nums[k	] - (nums[k-1]	+ difference)	) <epsilon))< td=""><td></td><td>binding within - 1010write.</td></epsilon))<>		binding within - 1010write.
Next number ascending Previous number descending (di	fference) AND next number ascending (difference)		۲ asce }	ndingCounter					This is because if I
CHECKING: 4.9 with 5.0 Inext number not descending (d	lifference)		sm.add(p System.o	otentialfurt	herAscendingBe	yondThisStart+	"->"+nums[k])	; WRITE RANGE"):	following
This is counter at the moment: 2next number not descending (d previous number descending(dif	0 lifference) ference) AND/OR next number ascending (difference)		System.o complete	ut.println(" Ticker(poten	-1010Writing r tialfurtherAsc	ange: " + pote endingBeyondTh	ntialfurtherA isStart,Strin	scendingBeyondThi g.valueOf(nums[k]	D.
Next number ascending Previous number descending (di	fference) AND next number ascending (difference)		//hasWri	ttenRepeatNu	mber=true; //	late change		4.7f,4.8f,4.	9f,5.0f,5.0f,4.9f,4.8f
CHECKING: 5.0 with 5.0 REPEAT NUMBER			potentia	lfurtherAsce	ndingBeyondThi	sEnd="";			
23229U5ING -1010Writing range: 4.7-> 5.0	S STORED TO WRITE RANGE	1514	if (Stri {	ng.valueOf(n	ums[k+1]).equa	ls(potentialfu	rtherAscendin	gBeyondThisEnd))	K!=nums.length-2 Then there are no
CHECKING: 5.0 with 4.9 Descending sequence (difference	e)	TI	CKER: A(4	4)D(2)					[4.7->5.0, 5.0->4.8]
		[4	.7->5.0,	5.0->4	4.9]				
The best logic would be to keep tra	ck of the tickerCounter once it performs -1	010writing							
in the counter is still same once it re	acries new area of code, I will perform a w	nte							

Overleaf is the code that has been introduced:

<pre>sm.add(potentialfurtherAscendingBeyondThisStart+"-&gt;"+nums[k]); System.out.println("</pre>	+ ni ;						
I introduced this new code in the k=nums.length-2 section. Basically when it reaches here and it is rea write the range (on basis that no others have been written since the above). It also is more failproof	ady to						
acknowledging the tickerCounter>0 to ensure it can not enter here inadvertently It will write 50.0-<49	.9						
<pre>665 if ((potentialfurtherAscendingBeyondThisStart=="")</pre>							
666 && (potentialfurtherAscendingBeyondThisEnd=="")							
667 && (hasWrittenRepeatNumber)							
668 & (tickerCounter==tickerCounterOnPreviousRange)	&& (tickerCounter==tickerCounterOnPreviousRange)						
669 && tickerCounter>0)	&& tickerCounter≻0)						
670 - {							
671 start=String.valueOf(nums[k]);							
672 end=String.valueOf(nums[k+1]);							
673 System.out.println("1201021Writing range: " + start + "->	" + end);						
674 sm.add(start+"->" + end);	sm.add(start+"->" + end):						
675 completeTicker(start,end,k,lengthNums);	completeTicker(start.end.k.lengthNums):						
676 tickerCounterOnPreviousRange=0:							
677 break:							
678							

I will now try ChatGPT extract 3 again, all seems ok

Before I go through all my test cases again, I am quite keen to revisit those test cases that failed in last documentation due to A combinations of A with A-D....

[4./->5.0, 5.0->4.9]

I will see if they still exist, note these are ONLY issues with the ticker.

//small extract taken from second chatGPT extract. ok, ticker=fail (USE EXAMPLE)

//40.1f, 40.1f, 35.1f, 35.2f, 35.3f, 35.2f, 35.1f, 85.6f, 85.5f, 85.4f, 85.3f, 85.2f, 19.6f, 19.7f, 19.8f, 19.9f, 20.0f, 19.9f, 19.8f, 63.5f

//3.7f,3.8f,3.7f,3.9f,4.0f,3.5f,3.6f,3.7f,3.8f,3.7f,45.5f,45.4f,3f,3.2f,3.3f //ok, ticker=fail A (A-D)

//ISSUES THESE //Related to oversized or undersized ticker, it relates to transition and also standalone value occuring prior to it

//3.5f,3.6f,3.7f,3.8f, 3.7f, 3.5f, 3.6f,3.7f,3.8f, 3.7f,3.6f, 3.5f,3.6f,3.7f,3.8f //ok, ticker=fail (A-D) (A-D) A

//5.0f, 48.5f,28.6f, 28.5f, 28.4f, 28.3f, 28.2f, 3.5f,3.6f, 3.5f //ok, ticker=ok //issues D (A - too large) D

//28.6f, 28.5f, 28.4f, 28.3f, 28.2f, 3.5f, 3.6f, 3.5f //rule does not work DAD //issues ok, ticker=ok

//It appears that ascendingcounter SHOULD NOT BE INCREASED IN THE NEW RULE IN THE COMPLETETICKER METHOD WHERE

//ascendingcounter==1 and descending in both directions....

//29.9f,30.0f,28.9f,28.8f,28.9f,29.0f //no issues ADA //ok, ticker=ok

I am now observing ticker being incorrect for a situation not experienced before.

85.6f, 85.5f,85.4f,65.1f,65.0f, 19.6f, 19.7f, 19.8f, 1	19.9f, 20.0f, 19.9f, 19.8f, 63.5f
TICKER: D(3)D(2)A(4)-D(3)S [85.6-> 85.4, 65.1-> 65.0, 19.6->20.0, 20.0->19.8, 63.5]	issues, this has two descending before (A-D)
No issues, this has one descending before (A-D)	
85.6f, 85.5f, 85.4f, 19.6f, 19.7f, 19.8f, 19.9f,	20.0f, 19.9f, 19.8f, 63.5f
TICKER: D(3)A(5)-D(3)S [85.6-> 85.4, 19.6->20.0, 20.0->19.8, 63.5]	

I will aim for now to discard these and try to acknowledge failed ticker whilst I go through test cases....

I might not be able to identify all of them given long test arrays..

I found another test case failing, it is similar concept to the above with new variable:



This is another failed case which I remediated

As I suspected earlier, all the scenarios are related to dealing with k=nums.length-2



The only other scenarios are having descending summary range and ascending summary range before it, and there are no issues... I also tried repeat numbers which match with the descent starting at k=nums.length-2 and no issues. I also changed the last range to ascending and no issues.