:) I know what you mean about the clashes being stressful, but I find them much more fun than these very simplistic puzzles that I just absolutely can’t figure out. I know I could ask you, but it feels like I should be able to complete them myself, so I continue banging my head against it.

I got 1 more shortest, so I have one left before completing that quest, and then whatever comes after that. You said I’d have the last few completed already? That’ll be nice to complete a branch.

\*sigh\* Ok, I’ve had a lot of fun with the puzzle(s) I’m working on. I thought I disliked puzzles because there was always an edge case at the end that I couldn’t solve; but I spent ~4 hours the past two days, staying up till 2 am this morning solving this one. And I need your help as to why this isn’t working, if you don’t mind.

if (LETTERS.Contains(temp[j])){

//Console.WriteLine(temp);

int ind = LETTERS.LastIndexOf(temp[j]);

let[ind] = "\0";

string hold = "";

for (int k = 0; k < LETTERS.Length; k++){

if (let[k] != null){

hold += let[k];

}

else {

break;

}

}

}

LETTERS is a string of available letters, temp is the next word to check, let[] is an array of the same available letters (1 per slot). This section is supposed to take the tested letter and subtract it from the available list, reset for each tested word. That way it can’t use more than the list, unless there’s some function that I’m missing. It can have duplicate letters inside the list, but it can’t use more than the available list once. I.E. I’m solving the Scrabble puzzle, this is the last section.

hicquwh

When I have a WriteLine after the for loop checking the hold string, it shows that h??qu??

h??qu??, testing for the word which, meaning it still referenced the 7th index for the last h in which. I’m not sure why it’s doing that, as it should reference the first h in the string, since the last is now a null? I’m still working through it; considering using a new for loop and referencing the string directly, as maybe the array is doing something funky.

I’ve now emptied that part of the code, enough to check for word length <= 7 and that it can’t contain letters outside of the allowed field. It’s enough for the first 5 tests. I really thought I had it.

I had something going that correctly did everything I was describing, then I just tried to figure out what happened when it didn’t contain the next letter by going to SoloLearn.. And then everything broke, and I destroyed it. So I had something at some point, but no record of it.

And then I go to an easy loop puzzle just to calm down and finally finish the quest (Horse Racing) and they put 100000 horses that makes my code time out every time! I can’t do the math fast enough, literally!

just moved some stuff around, it’s fine now.

just kidding, two additional hidden cases that break something because he keeps TIMING ME OUT.

i should stop.

Timing out is super frustrating. Quick glance at your scrabble code, have you tried removing the break statement? I think that exits the loop before getting to the end instead of just continuing to the next k value.

Also,

int ind = LETTERS.LastIndexOf(temp[j]);

let[ind] = "\0";

I think you are checking the LETTERS array but changing the let array. So you put nulls in the let array, but then you check the LETTERS array there are zero bills.

Also also, I think you can just put null instead of "\0".

[Scrabble](https://www.codingame.com/ide/puzzle/scrabble)

Yeah, as I mentioned in my second to last paragraph, I made a bunch of changes but didn’t save anything, so I had something that output exactly what I needed, and I just needed a way to decipher when a letter was no longer available, so once I go through my current round of puzzles I’ll get back to it.

Is timing out a C# thing, that it takes so long to read all the characters? So if I used a shorter coding language it’d be fine? Quite literally all the math was is a for loop and testing it against the next one (for the really big case) which worked, but the math \*technically\* requires checking against all later horses, so I dunno.

Timing out is not a C# thing. At the beginning, the computer compiles the code down to binary, and then the character count doesn't even matter. Timing out just means we're doing too many loops and need to figure out some optimization.

These problems often annoy me. I'm like "Hey, in the real world I could do this and if it took thirty seconds, it would be just fine." But optimization is a good thing to learn about.

**For example**, if you're testing each horse against each other horse, you're doing 100,000 \* 100,000 checks. We need to figure out a way to not do checks we don't need. We could set j = i instead of j = 0, because we already did those checks. **If** we're just looking for the fastest horse, we can quit the inner loop as soon as we find a faster horse. If horse1 is faster than horse0, we can save 99,998 checks.

If you're having timeout issues, other people probably are, too, and you should check the forums for tips. I'll look up the problem later and see if I can help.

EDIT: Yes, I tried it out and I timed out when I tested every single combination, but I was able to do something really simple to go from 10,000,000,000 loops to 99,999 loops. Spend a couple minutes thinking of how you could change the array to require fewer checks, and if you’re still stuck check out the forums or ask me. :)

We aren't trying to find the fastest horse? \*Sorry! I thought you knew the problem.

We need to find the smallest difference between any two horses, and since they don't have to be in order, you have to check all of them. As far as I can tell, there's no way to skip something like that. The really large test case they give you is in descending order, which is why you can simplify the code, but they break some invisible cases, so I assume if you could do the check for really large cases, it'd be fine.

I mean.. unless you found a way to organize them into an array? That sounds like sorting 99,999 horses would not be faster. But it'd make the final test case really easy. I mean, unless you use Bogosort.

It was an example to show that for some problems you can cut down on unnecessary loops. Then I said “I’ll look up the problem later and see if I can help.” Then I tried it and added my edit, so I do know the problem.

You recognized a way to shorten your loop count and you gave up on it. If a sorted array takes 100,000 checks and an unsorted array takes 10,000,000,000 checks, sorting that array just might be worth it.

Google “c# sort array”.

Ok, so it's just a single, built in function. I was just used to videos that show sorting algorithms slowly working through the bar graph, and assumed that sorting would be a slow result, and would time out before I even got to the math, which looking back on it sounds kinda dumb considering how fast computers are.. I doubt I'll have access to the computer before we go, so I'll have to do it once I get home tonight.

\*I think I also thought I'd have to write my own loops to sort the list which'd take much longer. (\* just means I came back with an edit, and it hasn't been hours.)

EDIT: Alright, I’m still failing All Horses Tie and Horses in Disorder invisible quests.. I’ll be checking the forums to see what the heck’s going on.

And.. that’s so weird. One of them suggested writing the new value into an array, then outputting the Min value of that. And it flipping worked. I can’t even know what was going wrong.

08/23 - Jesse

Yeah, I know that it’s impossible to know when you’re just getting started, what built-in functions there are. Weird that you still had problems with it, and that it mysteriously worked. If you want, you can show me your code and I’ll try to see what was going wrong with it. I can also take another look at your Scrabble code if you want to post it.

Or you can just tell me what you’re working on in CodinGame next. UVU’s semester starts today, so it’ll be a few days before I get back to CodinGame, probably. Still have to work on my CSB bot.

|| - Cody

I'd be interested in posting my code. Probably a silly mess up in the final check, but yeah. In a few hours it'll be here.

Scrabble hasn't changed at all; I deleted my code for the letter check, and everything else is still enough to calculate scores from a full array. Just need to write that.

I cheated my way through the Shortest Clash quest via 1v1 with someone who didn't finish, completed the next one where you win three in a row, and… now I have to do it in 3 different languages, so I guess I'm learning Python and.. what do you recommend? There's some random achievements for Bash and likely a few others (as well as solving 15 puzzles in every language imaginable :( ) but any others we could possibly use for game-making? Blender can read Python, I think.. yeah. Let me know.

Good luck with school! I have a week to catch up, I guess. Writing it down now; my plans are to finish Scrabble and work on the Mars Lander puzzle (Optimization #2?), which will also get me the next Puzzle one by completing two Medium puzzles. Then I'll move to some Bots; likely CSB. With some Clashes tossed in.

\*Oh, and I'll be throwing spaghetti at the Brainstorm Organized doc for "Field" just to try and fill it out.

class Solution

{

static void Main(string[] args)

{

int N = int.Parse(Console.ReadLine());

int[] array = new int[N];

for (int i = 0; i < N; i++)

{

int pi = int.Parse(Console.ReadLine());

array[i] = pi;

}

Array.Sort(array);

int final = 1000;

int[] test = new int[N-1];

for (int i = 0; i < array.Length; i++){

if (i == array.Length-1){

break;

}

//Everything above this works as intended.

else {

int test2 = Math.Abs(array[i] - array[i+1]);

test[i] = test2;

/\*if (test2 == 0){

break;

}

if (final > test2){

final = test2;

}\*/

}

}

Console.WriteLine(test.Min());

}

}

It’s not cheating if you get lucky with an easy opponent.

I would definitely recommend Javascript. It’s very lightweight and easy to use. Not bad for Shortest clashes, either (but will usually be beaten by Python or Ruby). It’s my programming language of choice for CodinGame.

Is that code the stuff that finally worked? Or what you had before you checked the forums? I just need to understand what question I need to be answering. :)

Yes, this is my final code. My first test is in comment brackets at the bottom of the for loop, and I just added int[] test in the middle with a new line of code in the end for loop, so the array output is what I made after reading the forums. So after forums, +1 arrays.

END OF FILE