**Problem 2. Hunting Rabbits**

## (a)

**Solution.** $C = 2, 3, 4, 5$

We want to find all such $C$'s.

Obviously, if $C = k$ is possible, so is $C = k + 1$. Thus, if we can find the **smallest** $C$, we could find **all** $C$'s.

To find the smallest $C$, we'll test $C = 1$ to $C = 5$ one by one (before we find a better way to do it).

Let $c(x)$ be a function that assigns each integer from 1 to 6 to a color denoted by $c_i$ such that $1 \leq i \leq C$.

When $C = 1$, it is obviously impossible to shoot the rabbit.

The simplest way to show this is that there is a **local pattern** that make hunting down the rabbit impossible: $c(k) = c(k+1) = c_i$. If the tracking device always reports $c_i$, $h_t$ could be either $k$ or $k+1$ for all $t$'s. So regardless of the hunter's strategy, the rabbit could always evade the dart.

When $C = 2$, let $c_1 = a$ and $c_2 = b$. To avoid coloring two consecutive integers the same color, the only coloring we could do is **ababab**.

With this coloring, it is easy to show that the rabbit could only be shot if it is 'trapped at 1 or 6'. So we came up with the following strategy to gradually **'close in from the left'**.

**Claim.** *The hunter could use the following strategy to hunt down the rabbit. Without loss of generality, we assume that the rabbit want to start on color b. Otherwise, reflect the entire operation below across point* 3.5 *on the number line.*

- *Shoot* 2 *when* $t = 1$. *Otherwise,*
  - *If the tracking device shows the same color as the previous round, shoot 2 units to the right of the previous shot.*
  - *Otherwise, shoot 1 units to the right of the position of the first shot in the previous streak of shots of the opposite color.*

*Proof.* The rabbit, knowing the hunter's strategy, would be at 4 or 6 at $t = 1$. If it stays in that position, it would be shot when $t = 2$ or $t = 3$. So at some point, it would hop onto color $a$.

When the rabbit moves to $a$ at $t = k$, the hunter would shoot one unit to the right of the first shot of the previous streak of shots on color $b$, meaning that it would shoot at 3.

Now, the only possible position for the rabbit at $t = k$ is 5. It would be shot at 3 and being at 1 would require it to be at 2 when $t = k - 1$, which is impossible because 2 was shot when $t = 1$ and the rabbit did not move from $t = 1$ to $t = k - 1$.

For the next step, if the rabbit stays on 5, or move to 4, it will get shot according to the hunter's strategy. So it will jump to 6 while the hunter shoot at 4.

Now, the rabbit is trapped! Either staying at 6 or jumping to 5 would get it shot. □

**Remark.** *This strategy could be used on a finite number line of arbitrary length.*

## (b)

**Solution.** $C = 3$ works.

Let the colors be $c_1$, $c_2$, $c_3$. For integer $n$, color the integer with $c_i$ if and only if $n \equiv i \pmod 3$.

Obviously, the hunter now knows whether the rabbit jumps left, right, or stays in the same position for each step. So the hunter knows the rabbit's total **displacement** $d$ from the start point at any time.

The set of possible starting points, a congruence class mod 3, is obviously countable. So the hunter could simply try them one by one. For example, if the potential starting points are equivalent to 0 mod 3, the hunter could test the starting points in the order $(0, 3, -3, 6, -6, 9 \cdots)$, meaning that the hunter would shoot at $(d, d+3, d-3, d+6, d-6, d+9 \cdots)$. And the hunter is guaranteed to shoot the rabbit after an **arbitrarily large** $t$.

**Remark.** *An arbitrarily large t is certainly a longggggg time to hunt down a rabbit, right? So one naturally ask: is there a strategy that guarantees shooting the rabbit within a certain time?*

*It turns out to be impossible. If we are coloring with a finite $C$, one color has to be applied to infinitely many integers. Say the rabbit decides to stay on one of the integers with that color for an arbitrarily long time. In the meantime, even in the simplest scenario where the integers with that color contain no consecutive pairs, the hunter would have no choice but to try all the integers with that color one by one, which would take an arbitrarily long time.*

**Remark.** *Another question that naturally arises is whether $C = 2$ is possible. It is easy to show that the rabbit coud always evade the hunter if the number line is colored with alternating colors as in (a). However, given that no integer could be visited by the rabbit infinitely many times, assigning consecutive pairs of integers the same color might no longer be problematic. I haven't investigated whether a $C = 2$ coloring based on this is possible.*

## (c)

**Solution.** $C = 6$ works.

**Remark.** *This part is inspired by Dragomir Grozev's blog linked in the question.*

We see each color as a tuple $(c_x, c_y)$.

For the $c_x$ layer, we use the same coloring as that in (b), going $\cdots c_3, c_1, c_2, c_3, c_1 \cdots$. This layer gives us the rabbit's total displacement $d$ at any time.

The $c_y$ layer is inspired by Grozev's third layer. While strips worked great, we are going to use blocks instead because they are slightly more direct.

We use black and white for this layer. We color the integer 1 black, the next 2 integers (2 and 3) white, the next 3 integers after that black, the next 4 integers after that white $\cdots$ And we color negative number $-a$ the opposite color of $a$.

If the rabbit always stays within 2 consecutive blocks of integers (for example, a black streak and a white streak), which contains a finite number of integers, one integer has to be visited infinitely many times due to the pigeonhole principle. This is not allowed. Thus, the rabbit must pass through at least 3 consecutive blocks of integers, which contains the segment $c_1, c_2 \cdots c_2, c_1$ in positional order. Note that the rabbit does not have to pass through the integers in this time order since we can figure out the direction of each jump from the

previous layer which helps us determine the positional order of the squares. Now the hunter could identify the exact position position of the path since each such segment with the unique number of $c_2$ in between the $c_1$'s appears on the number line exactly once.

So we need 3 colors for the $c_x$ layer and 2 colors for the $c_y$ layer, resulting in 6 tuples.

**Remark.** *I've tried to use a similar approach to come up with a 2-layer coloring method for the 2d case in Grozev's blog, but haven't gotten anything yet. I went slightly over the 1-hour suggested think time. I wish I could spend more time on it!*