Vision – Analysys

Author: Emil Indzhev

1 1 dimension

1.1 Naive solution

The easiest way to solve the problem is to think of such a pattern where we always know which way is left and which way is right. The simplest such pattern is $1, 2, 3$ or any variation (we'll use $3, 1, 2$). With such a pattern, we can always infer the correct direction by simply looking at the adjacent 2 sectors. It has an average vision of 2.0.

1.2 Full score using a 3

Various improvements are possible by just adding more 1s, such as $3, 1, 1, 2$ or $3, 1, 2, 1$ or even $3, 1, 1, 2, 1$. In all of these, all sectors know which way is left and which is right and can thus move the ship right only. However, to get to an average of 1.5 using this base idea, we need to notice that it is okay to have three 1s in a row since: the 3 can jump to the last one (and it knows the direction due to the other side where it sees the 2) and the middle 1 can just go to either one of the other 1s and they can go to their adjacent non-1 sector (instead of just going right). That way we can get to $3, 1, 1, 1, 2, 1$, which has an average of 1.5. We cannot add another 1 to the end since then the 2 will also not know which way to go.

1.3 Full score without using a 3

Another idea instead of having three $1s$ in a row after a 3 is to just replace the 3 with two 2s in the $3, 1, 1, 2, 1$ pattern: $2, 2, 1, 1, 2, 1$. Here all 2s know which way is left and right and can jump to the next 2 to the right (or to a 1 right before it), while all 1s can just jump to an adjacent 2. Thus, this is another solution with an average of 1.5. Furthermore, unlike the previous one, it also works backwards (so it is actually two different solutions), which we can write as: $2, 2, 1, 2, 1, 1$ (once again we cannot have two 1s in both slots).

1.4 Longer patterns

Some longer full score patterns can be constructed by combining elements of both of these: $3, 1, 1, 1, 2, 1, 2, 1$ or even $3, 1, 1, 1, 2, 1, 2, 2, 1, 1$.

1.5 On using 4s and larger numbers

It seems useless to include a 4 in the pattern since all it can do is jump to the last one of four 1s, but then the middle two 1s would get stuck in a cycle between each other. One other idea would be to use an even larger number to jump over a "confused" 2, something like: $5, 1, 1, 2, 1, 1, 2, 1$. However, this does not lead to a good average vision.

1.6 Optimality

The target of 1.5 was shown optimal using a bruteforce using numbers up to 3 (which seems justified due to the informal arguments for why larger numbers are not useful).

2 2 dimensions

After the naive solution, we'll examine various solutions based on the 1D solution. Then we'll look at custom ideas for the 2D case.

It is crucial to note that the key "idea" for getting the full (or even just a high) score for this subtask is that it is okay to have 1s surrounded by just 1s, as long as we have a consistent direction we can commit to (e.g. up/down, or left/right, or diagonal) from them, which always leads to a 1 adjacent to a non-1; we can call this an "exit" strategy.

It is cumbersome to explain the precise movement rules (such as exit strategy and how to detect which sector we're in and what the different orientations look like) for all patterns shown. You are strongly encouraged to take an active approach when examining the patterns shown in this section, trying to figure out how to traverse them – focus on marking "bad" 1s (surrounded by just 1s) and finding the valid exit strategy (left/right, or up/down, or diagonal), then make sure you see how all non-1s can work.

2.1 Naive solution

Here, the most naive solution is perhaps one where we can easily tell which sector we're in and which way is which based on adjacent sectors (many different movement rules can traverse this pattern):

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

2.2 Rainbow patterns

A somewhat better idea is to make both every row and every column be an instance of a valid 1D solution. Then to traverse this, we can separately figure out a horizontal and a vertical move and either do a combination of the two (this has some problems for more complex 1D patterns) or just sometimes do the horizontal one and other times the vertical one (e.g. vertical move on a 3, horizontal move otherwise). Such patterns have the same average vision as their 1D base patterns, which is not very good. Here are 2 of them:

$$\begin{bmatrix} 3 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 & 1 & 3 \\ 1 & 1 & 2 & 1 & 3 & 1 \\ 1 & 2 & 1 & 3 & 1 & 1 \\ 2 & 1 & 3 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 & 1 & 2 \end{bmatrix}$$

2.3 Spacing-based patterns

Another idea is to just have some rows be valid 1D solutions but separate them using "empty" (all 1) rows. We can move horizontally using the 1D solution and we figure out how to move vertically using the gaps. Another consideration is to make sure that if we start at a 1 in a 1-only row, we move to an adjacent non-1 number (even if on a different row). For some solutions, we may even have 1s surrounded by just 1s; different patterns require different exit strategies (and some have no valid exit strategy and thus are not traversable).

A solution of this type based on 3, 1, 1, 1, 2, 1 using gaps of 0, 1 and 2 is possible (with an average vision of 1.25). We can do just horizontal moves on 2s and a combined vertical and horizontal move on a 3.

$$\begin{bmatrix} 3 & 1 & 1 & 1 & 2 & 1 \\ 3 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

We can improve this to have gaps of 1, 2 and 3. This needs the last jump to be done to the adjacent 1 above the "target". However, the issue is that we cannot have both a gap of 3 vertically and three 1s in a row in the horizontal pattern, so we switch it to $3, 1, 1, 2, 1$, getting an average of 1.2:

$$\begin{bmatrix} 3 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

We can improve this further, by making the rows more efficient. We still need to have a 3, but not be the best solution with a 3, so we can just use this followed by a bunch of copies (as many as we can) of a solution without a 3, i.e. $3, 1, (1, 2, 1, 2, 2, 1)$, which can get an average of 1.17262 with nine repetitions.

2.4 Offset-based patterns

The issue with the previous approach is that we don't leave the maximum number of empty rows, due to using the gaps for navigation. A better idea is to always have 2 (or even 3) empty rows between each instance of the 1D solution. Then, we can have each such instance offset in a different way, in order to recognize which instance is the up one and which one is the down one (notice that when doing this, we've already established the left/right direction). Here, again, we'll use a pattern with at least one 3 or a 4 (in order to support a gap of 2 or 3, respectively), where we'll do a combined horizontal and vertical move.

Here is a pattern based on $3, 1, 1, 1, 2, 1$ with a gap of 2 an average of 1.16667 (the down direction is always the one where the 3 is to the right of the current 3):

$$\begin{bmatrix} 3 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 3 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

An alternative is to include a 4, make the pattern a bit longer and use a gap of 3 to again get an average of 1.16667 (here the offsets are 0, 1 and -1, so we can again tell which way down is):

$$\begin{bmatrix} 4 & 1 & 1 & 2 & 1 & 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 & 2 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 2 & 2 & 1 & 1 & 4 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Once again, we can repeat the last part of the pattern $4, 1, 1, (2, 1, 2, 2, 1, 1)$ up to nine times to get an average of 1.13158. In fact, if it weren't for the limit of $N, M \leq 60$, we could repeat this an arbitrary number of times and, in the limit, get to 1.125.

2.5 Diagonal patterns – full score

Another idea is to put a valid 1D solution diagonally with empty diagonals separating it from the next instance of the solution and so on. The advantages of this are two: first, we only need to go down the same diagonal (and not "alternate" horizontal and vertical moves); second, a diagonal layout leaves fewer 1s surrounded by just 1s, allowing for easier/larger gaps.

One such solution based on the $3, 1, 1, 1, 2, 1$ solution with gaps of 2 has an average of 1.16667:

$$\begin{bmatrix} 3 & 1 & 1 & 3 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

A better idea is to have gaps of 3. However, this doesn't work if the base solution has three 1s in a row, so we can instead use the $2, 2, 1, 1, 2, 1$ solution to get an average of 1.125:

$$\begin{bmatrix} 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Interestingly enough, it is also possible to support gaps of 4, but the base pattern needs to not have even two consecutive 1s, which is too limiting to lead to a full score solution (but using $3, 1, 2, 1, 2$, we can get to an average of 1.16).

An alternative idea is to have much larger gaps but include some extra 2s in the empty regions to make sure that all 1s have an adjacent non-1 or an adjacent 1 with an adjacent non-1 (and in a predictable type of direction at that, i.e. a valid exit strategy). Here is such a solution with a single instance of the $3, 1, 1, 1, 2, 1, 2, 1$ pattern along the diagonal and just four extra 2s placed strategically in the empty regions to allow for valid exit strategies for the 1s (and to make sure the 2s can get to the diagonal); it's average is also 1.125:

$$\begin{bmatrix} 3 & 1 & 1 & 2 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 2 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

2.6 Custom patterns – full score

Another approach is to simply hand construct a valid 2D pattern instead of trying to generalize a 1D one. There is less to say here, since it is not as systematic as the other approaches described. Various different solutions are possible with generally high scores. The simplest full score solution (or, rather, solutions) use just three 2s in a 4 by 6 grid. One such solution is essentially just the $2, 2, 1, 2, 1, 1$ solution but also putting the 2s on descending rows (making sure each can see what it needs to and that there is a valid exit strategy for the 1s). This can also be looked at as a diagonal-like solution (but with a non-straight and non-one-to-one diagonal). It also has an average vision of 1.125:

$$\begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

2.7 Optimality

2D is too large a space to bruteforce properly and we have no proof of optimality. However, trial and error and the fact that distinct approaches all cap out at 1.125, leads us to believe that 1.125 is indeed the minimum possible average vision.

3 3 dimensions and up

No exploration has been done on higher dimensions, but feel free to explore it at your leisure. Our conjecture is that the optimum score tends to 1.0 in the limit.