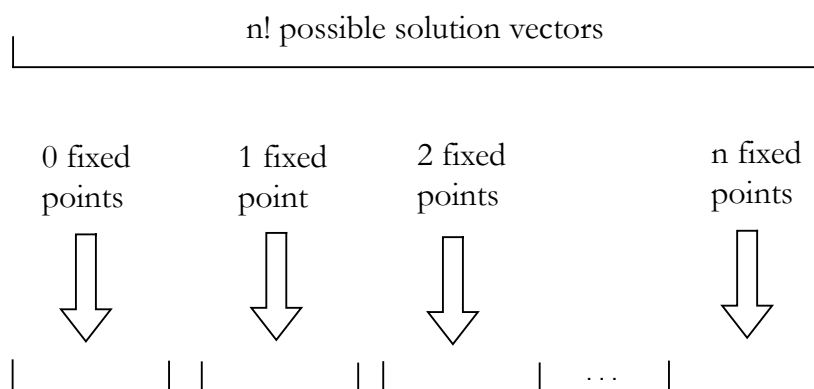


## Mastermind with No Repeats: Lower Bound for Guessing Strategies

1. *Game.* In this version of Mastermind, there are  $n$  spots and  $n$  colors; all guesses and solutions are permutations.
2. *Trivial Lower Bound.* We have  $n!$  possible hidden vectors, and each query has  $n$  possible responses.<sup>1</sup> Since we must distinguish between every pair of vectors, we must ask at least

$$\log_n(n!) \sim \left( n - \frac{n}{\ln(n)} \right)$$

3. *Buckets.* In order to beat the trivial lower bound, we use a more accurate picture of the distribution of the possible solutions after feedback from a given guess. For any guess, we can split the possible solutions up into groups, which we will call *buckets*; one bucket for each possible response given by the codemaker. For any guess we make, the number of possible solutions that give the response “ $k$  (black) hits” is the number of permutations with  $k$  fixed points with respect to our guess vector.



The size of a bucket of possible solutions with  $k$  hits can be counted easily: we choose the  $k$  colors that are fixed points (wrt our guess vector), then we permute the remaining  $n - k$  colors without any fixed points, giving  $D(n - k)$ , the derangements on an  $(n - k)$ -element vector:

$$\binom{n}{k} D(n - k)$$

This tells us the initial size of each bucket, and as more guesses are made, the size of each bucket can only decrease, no matter which guess is considered.

4. *Notation.* Let  $S_n(t)$  be the minimum number of possible solutions remaining after  $t$  guesses for any guessing strategy. We compute this lower bound by using a set of guesses which are guaranteed to eliminate at least as many solutions as any other guess we could make. The worst-case scenario for this set of guesses therefore

---

<sup>1</sup>The possible responses are  $0, 1, \dots, n - 2, n$ , since it is impossible to get  $n - 1$  black hits with no repeats and  $n = k$ .

provides a lower bound for the number of guesses required by any guessing strategy to guarantee finding the hidden vector.

### 5. Recurrence Relation on Number of Possible Solutions after $t$ Turns:

- (a) We will inductively prove a lower bound,  $S_n(t)$ . By the discussion above, we know the maximum size of each bucket containing possible solutions. The best possible guess distributes the remaining solutions as evenly as possible so that the bucket containing the most remaining solutions is as small as possible. This best-possible distribution will fill some number of smaller buckets entirely and split the remaining solutions evenly into some number  $x$  larger buckets. That is, there are  $x$  incompletely filled buckets.
- (b) At the beginning of the  $t$ -th turn, we have at least  $S_n(t-1)$  possible solutions remaining. As we showed before, the size of each bucket is bounded by its original size of  $\binom{n}{k}D(n-k)$ . Thus the number of solutions remaining  $(t-1)$ -th turn relates to the value of  $S_n(t)$  as follows:

Let  $B_i(t)$  be the exact number of possible solutions remaining in the bucket holding possible solutions with  $i$  fixed points during the  $t$ -th turn (after submitting the guess vector, but before feedback). So

$$\begin{aligned}
 S_n(t-1) &= \sum_{i=0}^n B_i(t) \\
 &= \sum_{i=0}^{x-1} B_i(t) + \sum_{i=x}^n B_i(t) \\
 &\leq \sum_{i=0}^{x-1} S_n(t) + \sum_{i=x}^n \binom{n}{i} D(n-i) \quad \text{for optimal } x \\
 &= x \cdot S_n(t) + \sum_{i=x}^n \binom{n}{i} D(n-i)
 \end{aligned}$$

Once we know  $S_n(t)$ , picking  $x$  too large results in some smaller boxes overfilled. Picking  $x$  too small results in some larger boxes overfilled. So changing  $x$  to be something other than optimal always increases the RHS of this equation. Thus we have

$$S_n(t-1) \leq x \cdot S_n(t) + \frac{n!}{x!} \quad \forall x$$

Where  $t$  is the number of turns taken,  $S_n(t)$  is the number of possible solutions remaining after the  $t$ -th turn, and  $D(k)$  is the derangements on a set of size  $k$ .

### 6. Claim.

$$\sum_{i=x}^n \binom{n}{i} D(n-i) \leq \frac{n!}{x!}$$

This inequality leads to our bound on  $S_n(t)$  given below.<sup>2</sup>

*Proof.*

$$\begin{aligned}
 \sum_{i=x}^n \binom{n}{i} D(n-i) &= \frac{n!}{x!} \sum_{i=x}^n \frac{x!}{i!(n-i)!} D(n-i) \\
 &\quad \text{we use the fact that } \frac{x!}{i!} \leq \frac{1}{(i-x)!} \\
 &\leq \frac{n!}{x!} \sum_{i=x}^n \frac{1}{(i-x)!(n-i)!} D(n-i) \\
 &\quad \text{let } k = i - x \\
 &= \frac{n!}{x!} \sum_{k=0}^{n-x} \frac{1}{k!(n-x-k)!} D(n-x-k) \\
 &= \frac{n!}{x!(n-x)!} \sum_{k=0}^{n-x} \binom{n-x}{k} D(n-x-k) \\
 &\quad \text{note the summation counts all possible permutations} \\
 &= \frac{n!}{x!(n-x)!} (n-x)! \\
 &= \frac{n!}{x!}
 \end{aligned}$$

□

(a) Plug the above result into the inequality to get

$$S_n(t-1) \leq x \cdot S_n(t) + \frac{n!}{x!}$$

So we have

$$S_n(t) \geq \frac{1}{x} \left( S_n(t-1) - \frac{n!}{x!} \right)$$

---

<sup>2</sup>A combinatorial argument can be made as follows: The LHS denotes the number of permutations of an  $n$ -element vector which have at least  $x$  fixed points. Suppose we choose  $x$  fixed points and simply permute the rest of the vector. This gives  $\binom{n}{x}(n-x)! = \frac{n!}{x!}$  possible permutations. Clearly this includes all vectors with at least  $x$  fixed points (and overcounts by some margin), so the inequality holds.

**Theorem 1.** *For all  $n$ , and for any constant  $C_n$ , we have*

$$\frac{S_n(t)}{n!} \geq \frac{C_n! - (H_{C_n+t} - H_{C_n})}{(C_n + t)!}$$

$$\text{where } 0 \leq t \leq n - C_n$$

Here we use  $H_n$  to denote the  $n$ -th harmonic number.<sup>3</sup>

*Proof.* While this equation holds for all  $x$ , to achieve a desirable bound, we will let  $x = t + C_n$  for any positive constant  $C_n = C_n(n)$ . We have

*Base Case.* When  $t = 0$ , we have  $S_n(0) = n!$  and

$$\frac{S_n(0)}{n!} \geq \frac{C_n! - (H_{C_n} - H_{C_n})}{C_n!} = 1$$

*Inductive Step.* Assume the induction hypothesis holds for  $t - 1$ .

$$\frac{S_n(t)}{n!} \geq \frac{1}{x} \left( \frac{S_n(t-1)}{n!} - \frac{1}{x!} \right)$$

Let  $x = C_n + t$ .

$$\frac{S_n(t)}{n!} \geq \frac{1}{C_n + t} \left( \frac{S_n(t-1)}{n!} - \frac{1}{(C_n + t)!} \right)$$

Inductively,

$$\begin{aligned} \frac{S_n(t)}{n!} &\geq \frac{1}{C_n + t} \left( \frac{C_n! - (H_{C_n+t} - H_{C_n})}{(C_n + t - 1)!} - \frac{1}{(C_n + t)!} \right) \\ &\geq \left( \frac{C_n! - (H_{C_n+t-1} - H_{C_n}) - \frac{1}{C_n+t}}{(C_n + t)!} \right) \\ &\geq \left( \frac{C_n! - (H_{C_n+t} - H_{C_n})}{(C_n + t)!} \right) \end{aligned}$$

Then

$$\begin{aligned} S_n(n - C_n) &\geq n! \left( \frac{C_n! - (H_n - H_{C_n})}{n!} \right) \\ &\geq C_n! - (H_n - H_{C_n}) \end{aligned}$$

Thus the inequality holds for all  $n$ .

□

---

<sup>3</sup> $H_n = \sum_{i=1}^n \frac{1}{i}$

We choose  $C_n$  by the following reasoning: we want to achieve a lower bound of  $n - C_n$  turns. Thus, we want  $S_n(n - C_n) > 1$ , which follows if

$$C_n! - (H_n - H_{C_n}) > 1$$

For  $C_n = \log \log n$ , this is  $\geq 1$  for  $n \geq e^{198}$ .

For  $C_n = \log n$ , this is true for  $n \geq 10$ .

In conclusion, when  $n$  is sufficiently large, the minimum number of remaining possible solutions after  $n - \log \log n$  guesses is at least

$$\begin{aligned} S_n(n - \log \log n) &\geq (\log \log n)! - (H_n - H_{\log \log n}) \\ &> 1 \end{aligned}$$

Thus there is no strategy that can reduce the set of possible solutions to one, and guarantee finding the codemaker's hidden vector, in  $n - \log \log n$  turns or fewer.

## Lower Bound on Guessing Effectiveness for Permutation Game

**Theorem 2.** *In Mastermind with no repeats with  $n$  spots and  $k$  colors, for any set of possible remaining solutions, there exists a guess vector for which any response will eliminate at least  $1/nk$  of the remaining solutions.*

1. *Notation.*

- i. *Basics.* Let  $n$  be the number of spots in the hidden vector. Let  $k$  be the number of colors available. Since we allow no repeats in this variant of Mastermind, we have  $k \geq n$ . We will use single-count responses, which are the number of spots in which our guess vector and the hidden vector have the same color in that spot.
- ii. *Remaining Solution Set.* As the game proceeds, we maintain a set  $S$  that contains all possible solution vectors that haven't been ruled out by a previous guess and response.
- iii. *Bucket.* As defined in the previous section, a guess splits the remaining possible solutions into "buckets," where a bucket  $B_i$  contains all the remaining possible solutions with  $i$  fixed points with respect to the guess vector.
- iv. *Sub-Bucket.* Within a bucket, we will divide the vectors based on a chosen component, say the first element of each vector. Then there will be  $k$  sub-buckets  $D_i$ , where  $D_1$  contains all vectors from the bucket with a 1 in the first spot,  $D_2$  contains all those with a 2 in the first spot, and so on.

2. *Proof.* We give a proof by induction on  $n$ .

*Base Case.* When  $n = 1$ , we have a single-element vector. So there are at most  $k$  possible solutions remaining at any step, thus  $|S| \leq k$ . If we guess one of the remaining solutions  $v \in S$ , this will eliminate 1 solution ( $v$  itself) if the response is 0 and  $|S| - 1$  solutions if the response is 1. So we're guaranteed to eliminate at least  $1/|S| \geq 1/(nk)$  solutions.

*Inductive Step.* Assume that for  $n - 1$  spots and any number of colors  $\hat{k}$  (where  $n \geq 2$ ), we are guaranteed that there is a guess that eliminates a fraction at least  $1/(n\hat{k})$  of the solutions.

Consider a guess vector  $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ . We will analyze the number of solutions eliminated from  $S$  by guessing  $\sigma$  in the worst-case. We divide  $S$  into  $n$  buckets  $B_1, B_2, \dots, B_n$ , where  $B_i$  is as defined above ( $i$  fixed points with respect to  $\sigma$ ). Consider the response  $j$  that eliminates as few possible solutions as possible from  $S$ , so  $|B_j| = \max\{|B_1|, |B_2|, \dots, |B_n|\}$ .

Suppose that

$$|B_j| \leq \frac{nk - 1}{nk} \cdot |S|$$

Then the response  $j$  would eliminate at least  $|S|/(nk)$  solutions from  $S$ , and since  $j$  eliminates the fewest possible solutions of any response, all responses to  $\sigma$  eliminate a fraction at least  $1/(nk)$  of the possible solutions in  $S$ .

Otherwise we have

$$|B_j| > \frac{nk-1}{nk} \cdot |S|$$

*Case 1.* If  $j > 0$ , the total number of hits among all vectors in  $B_j$  is  $j \cdot |B_j|$ . These hits are divided up among  $n$  spots, therefore by the pigeonhole principle, at least one position must have at least

$$\frac{j}{n} \cdot |B_j|$$

Vectors that match the guess at that spot. WLOG, we assume this is position one. We now divide the bucket  $B_j$  into sub-buckets  $D_i$ , where permutation  $s \in B_j$  is put into sub-bucket  $D_{s_1}$ . From above, we have

$$|D_{\sigma_1}| \geq \frac{j}{n} \cdot |B_j|$$

We have two sub-cases:

i.  $|D_{\sigma_1}| \leq \frac{n-1}{n} \cdot |B_j|$ :

Then we apply the pigeonhole principle on the remaining solutions in  $B_j$  to find the second-largest sub-bucket: By our assumption, there are at least  $|B_j|/n$  solutions remaining in  $B_j$ , and there are  $k-1$  remaining sub-buckets  $D_i$ . By the pigeonhole principle, there exists a  $D_\alpha$  such that

$$|D_\alpha| \geq \frac{1}{n} \cdot \frac{1}{k-1} \cdot |B_j|$$

If  $\alpha$  is an element of  $\sigma$ , then WLOG let the color be  $\sigma_2$ . Then in this case we consider the guess vector  $\bar{\sigma} = \{\sigma_2, \sigma_1, \sigma_3, \dots, \sigma_n\}$ . Otherwise, consider the guess vector  $\bar{\sigma} = \{\alpha, \sigma_2, \sigma_3, \dots, \sigma_n\}$ . Consider the buckets  $C_i$  formed from the guess vector  $\bar{\sigma}$ . In either case, the vectors in  $D_{\sigma_1}$  will have at least one fewer hit with  $\bar{\sigma}$  than with  $\sigma$ , and the guess vectors in  $D_\alpha$  will have at least one more hit. Therefore, any response to  $\bar{\sigma}$  which corresponds to some vectors in  $D_{\sigma_1}$  corresponds to none of the vectors in  $D_\alpha$  and vice versa. So we know that any response to  $\bar{\sigma}$  will entirely eliminate at least one of those two sets. From above, we know

$$\begin{aligned} |D_{\sigma_1}| &\geq \frac{j}{n} \cdot |B_j| \\ &\geq \frac{j}{n} \cdot \frac{nk-1}{nk} \cdot |S| \\ &\geq \frac{1}{n} \cdot \frac{n}{nk} \cdot |S| \\ &= \frac{1}{nk} \cdot |S| \end{aligned}$$

And

$$\begin{aligned} |D_\alpha| &\geq \frac{1}{n} \cdot \frac{1}{k-1} \cdot |B_j| \\ &\geq \frac{1}{nk-n} \cdot \frac{nk-1}{nk} \cdot |S| \\ &\geq \frac{1}{nk} \cdot |S| \end{aligned}$$

Since  $\bar{\sigma}$  will always eliminate at least one of these two sets, we also have that  $\bar{\sigma}$  will eliminate at least  $|S|/(nk)$  vectors from  $S$  with any response.

ii.  $|D_{\sigma_1}| > \frac{n-1}{n} \cdot |B_j|$ :

Consider the set of vectors in  $D_{\sigma_1}$ . By definition, all of them have  $\sigma_1$  as their first entry. Their remaining  $n-1$  entries are made up of the remaining  $k-1$  colors, and no two of them coincide in all spots  $2, 3, \dots, n$  (otherwise they would be the same vector). Therefore, if we consider this new set of  $n-1$  vectors formed by taking spots  $2, 3, \dots, n$  of each vector in  $D_{\sigma_1}$ , they are a set of possible solutions for this Mastermind game with  $n-1$  spots and  $k-1$  colors. By the induction hypothesis, there is a guess of size  $n-1$  that is guaranteed to eliminate  $1/((n-1)(k-1))$  of these vectors. If we add  $\sigma_1$  to the beginning of this guess, it becomes a guess of size  $n$ . For any response  $r$ , any of the vectors in  $D_{\sigma_1}$  will be eliminated if and only if their vector of spots  $2, 3, \dots, n$  was eliminated by the response  $r-1$  in the smaller game of Mastermind. Therefore, this guess still guarantees that we eliminate

$$\begin{aligned} \frac{1}{(n-1)(k-1)} |D_{\sigma_1}| &\geq \frac{1}{(n-1)(k-1)} \cdot \frac{n-1}{n} \cdot |B_j| \\ &\geq \frac{1}{(k-1)} \cdot \frac{1}{n} \cdot \frac{nk-1}{nk} \cdot |S| \\ &\geq \frac{1}{nk-n} \cdot \frac{nk-1}{nk} \cdot |S| \\ &\geq \frac{1}{nk} \cdot |S| \end{aligned}$$

*Case 2.* If  $j = 0$ , we consider the first component of all vectors in the bucket  $B_0$ . Since  $j = 0$  and  $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ , no vector in  $B_0$  can contain  $\sigma_1$  in the first position, so there are  $k-1$  possible colors in the first position. By the pigeonhole principle, there must be one color which appears in the first spot of at least

$$\frac{1}{k-1} \cdot |B_0|$$

vectors in  $B_0$ . If this color appears in  $\sigma$ , WLOG let the color be  $\sigma_2$ . Then in this case we consider the guess vector  $\bar{\sigma} = \{\sigma_2, \sigma_1, \sigma_3, \dots, \sigma_n\}$ . Otherwise, call this new color  $\tau$  and consider the guess vector  $\bar{\sigma} = \{\tau, \sigma_2, \sigma_3, \dots, \sigma_n\}$ . In either case,  $\bar{\sigma}$  matches these  $|B_0|/(k-1)$  vectors in either one or two positions.



Consider the buckets  $C_i$  formed by the guess  $\bar{\sigma}$ . We know

$$\begin{aligned} |C_1| + |C_2| &\geq \frac{1}{k-1} \cdot |B_0| \\ &\geq \frac{1}{k-1} \cdot \frac{nk-1}{nk} \cdot |S| \\ &\geq \frac{1}{k-1} \cdot \frac{nk-n}{nk} \cdot |S| \\ &= \frac{n}{nk} \cdot |S| \\ &\geq \frac{2}{nk} \cdot |S| \end{aligned}$$

Therefore we have either

$$|C_1| \geq \frac{1}{nk} \cdot |S| \quad \text{or} \quad |C_2| \geq \frac{1}{nk} \cdot |S|$$

Let this larger bucket be  $C_\alpha$ , then suppose

$$|C_\alpha| \leq \frac{nk-1}{nk} |S|$$

Then getting the response  $\alpha$  will eliminate at least  $|S|/(nk)$  possible solutions from  $S$ , and getting any other response will eliminate at least  $|C_\alpha| \geq |S|/(nk)$  possible solutions. Therefore the guess  $\bar{\sigma}$  is guaranteed to eliminate at least  $|S|/(nk)$  solutions from  $S$ .

Otherwise we have

$$|C_\alpha| > \frac{nk-1}{nk} |S|$$

Then, this satisfies the conditions for Case 1 above, where we have a guess whose largest bucket is  $B_j$  with  $j > 0$ .

□

3. The minimax algorithm is a guessing strategy described by Knuth [DK76]. At each turn, the algorithm assigns each of the possible guesses a score equal to the minimum number of solutions that they would eliminate over all possible responses from the codemaker. It then picks the guess with the maximum score.

*Corollary.* The minimax algorithm on Mastermind with  $n$  spots,  $k$  colors, and no repeats takes at most

$$\frac{\log \left( \frac{k!}{(k-n)!} \right)}{\log \left( \frac{nk}{nk-1} \right)}$$

turns to find the hidden vector.

*Proof.* Each step in the minimax algorithm cuts the set of possible solutions  $S$  down by a factor of  $(nk - 1)/(nk)$  at each step. To complete the game, we need to cut  $S$  down by

$$\frac{1}{\left(\frac{k!}{(k-n)!}\right)}$$

Thus the required number of steps is

$$\log_{\left(\frac{nk-1}{nk}\right)} \left( \frac{1}{\frac{k!}{(k-n)!}} \right)$$

Which is equivalent to the claim. □

This gives us the following asymptotic bounds:

- i. The minimax algorithm for Mastermind with  $n$  spots,  $k$  colors, and no repeats takes at most  $O(n^2 k \log k)$  turns to find the hidden vector.
- ii. In the same game where  $n = k$ , the minimax algorithm takes at most  $O(n^3 \log n)$  turns to find the hidden vector.

Both of these asymptotic bounds are actually strict upper bounds, and each of them differs from the exact bound by a constant factor which approaches 1 as  $n$  approaches infinity.

## Extension of Previous Section to Repeated Colors

In order to replicate the bound of  $nk$  with repeated colors, we need to make the following assumption:<sup>4</sup> There exists a guess vector  $v$  such that all of the solutions in set  $S$  correspond to a single response  $r$  after making the guess  $v$ . Note that this is always satisfied if the solutions are any set of solutions remaining after any set of guesses in an actual game of Mastermind. We simply let the guess vector  $v$  be any guess vector that has already been guessed. Clearly, all solutions remaining correspond to whichever response was given to that guess vector, since guesses corresponding to any other response were eliminated. We will also prove the theorem for the special case of the very first guess separately.

**Theorem 3.** *In a game of Mastermind with  $n$  spots and  $k$  colors, for any set of remaining solutions that can actually be achieved during a game of Mastermind, there is always a guess for which any response will eliminate at least  $1/nk$  of the remaining solutions.*

*Proof.* We will follow the same logic as taken in the previous proof, but will describe only the parts that differ from the previous proof.

1. Because of the problem statement, we will separately prove the base case where  $S$  is the set of all  $k^n$  possible solutions before any guesses have been made. In this case, we guess the vector  $\{1, 1, \dots, 1\}$ . The bucket  $B_j$  has size

$$|B_j| = \binom{n}{j} (k-1)^{n-j}$$

So the greatest fraction of possible solutions remaining is

$$\max_j \left( \frac{\binom{n}{j} (k-1)^{n-j}}{k^n} \right)$$

When  $k \geq n$ , we have

$$\begin{aligned} \frac{\binom{n}{j} (k-1)^j}{k^n} &\leq \frac{n^{n-j} (k-1)^j}{k^n} \\ &\leq \frac{k^{n-j} (k-1)^j}{k^n} \\ &= \frac{(k-1)^j}{k^j} \\ &\leq \frac{k-1}{k} \end{aligned}$$

Therefore, we always eliminate a fraction at least  $1/k \geq 1/nk$  of the solutions.

---

<sup>4</sup>The proof in the previous section can be extended easily to general  $n$  and  $k$ : Substitute to show that there is always a guess that eliminates a factor of  $1/n^2k$  possible solutions at each step. For a tighter bound with repeated colors, however, we need to make more restrictive assumptions.

When  $k < n$ , it is difficult to find the size of one bucket compared to the entire set, so instead we will compare one bucket to the sum of itself and the next bucket. This ratio will always be at least the ratio of one bucket compared to the entire set.

$$\begin{aligned}
 \frac{|B_{j+1}|}{|S|} &\leq \frac{|B_{j+1}|}{|B_j| + |B_{j+1}|} \\
 &\leq \frac{1}{\frac{|B_j|}{|B_{j+1}|} + 1} \\
 &\leq \frac{1}{\frac{\binom{n}{j}(k-1)^{n-j}}{\binom{n}{j+1}(k-1)^{n-j-1}} + 1} \\
 &\leq \frac{1}{\frac{j+1}{n-j}(k-1) + 1}
 \end{aligned}$$

We plug in  $k \geq 2$  (the entire problem is trivial when  $k = 1$ , as there is only 1 solution).

$$\begin{aligned}
 \dots &\leq \frac{1}{\frac{j+1}{n-j} + 1} \\
 &\leq \frac{n-j}{n+1} \\
 &\leq \frac{n}{n+1} \\
 &= 1 - \frac{1}{n+1} \\
 &\leq 1 - \frac{1}{nk}
 \end{aligned}$$

Additionally, for bucket 0, we have:

$$\begin{aligned}
 \frac{|B_0|}{|S|} &= \frac{(k-1)^n}{k^n} \\
 &\leq \frac{k-1}{k} \\
 &\leq 1 - \frac{1}{nk}
 \end{aligned}$$

So we have that the first guess of any game will always eliminate at least  $1/nk$  of the remaining solutions.

2. The argument for the base case where  $n = 1$  proceeds exactly as in the previous section.
3. The induction hypothesis remains almost the same: Assume that for  $n - 1$  spots and any number of colors  $\widehat{k}$  (where  $n \geq 2$ ), we are guaranteed that there is a guess that eliminates a fraction at least  $1/(n\widehat{k})$  of the solutions, given that there is a guess such that all solutions fall into the same bucket.

4. *Main Proof.*

Let this bucket be  $j$ . From our assumptions  $B_j = S$ . Then we have two subcases as before:

*Case 1.* If  $j > 0$ , we make the following adjustments to the previous proof.

- i. For the first subcase, we are now allowed repeats, so we can always let

$$\bar{\sigma} = \{\alpha, \sigma_2, \sigma_3, \dots, \sigma_n\}$$

The proof of the rest of this subcase follows.

- ii. For the second subcase, we proceed as the previous section did by considering  $D_{\sigma_1}$ , but now we are no longer guaranteed that the spots 2 through  $n$  of these vectors have no instance of  $\sigma_1$  since colors can be repeated. Whereas before, inducted on  $n - 1$  spots with  $k - 1$  possible colors, we now have to induct on  $n - 1$  spots still with  $k$  possible colors. This is fine, though, as by the induction hypothesis we can therefore eliminate

$$\begin{aligned} \frac{1}{(n-1)k} \cdot |D_{\sigma_1}| &\geq \frac{1}{(n-1)k} \cdot \frac{n-1}{n} \cdot |B_j| \\ &= \frac{1}{nk} \cdot |S| \end{aligned}$$

It is important to check that the induction hypothesis holds- namely, that there exists a guess of size  $n - 1$  such that all of the  $(n - 1)$ -vectors we're considering produce the same response.

However, by our assumptions,  $\sigma$  is an  $n$ -vector such that all of the  $n$  vectors fall into a single bucket  $j$ . Since every vector we're considering for induction is in the same sub-bucket  $D_{\sigma_1}$ , if we consider only spots 2 through  $n$ , all of these  $(n - 1)$ -vectors will have exactly one fewer hit than they did as  $n$ -vectors, (as all of them had a hit in the first spot), so all of them fall into the same bucket (bucket  $j - 1$ ) for the guess  $\{\sigma_2, \sigma_3, \dots, \sigma_n\}$ , so the condition is satisfied.

*Case 2.* If  $j = 0$ , we can no longer find a new  $\bar{\sigma}$  and switch to Case 1, so instead we induct similarly to Case 1. We split the bucket up into sub-buckets as before. By pigeonhole principle, there must exist an  $\alpha$  such that  $|D_\alpha| \geq |B_j|/(k-1) \geq |S|/(nk)$  (since  $j = 0$  we have no hits, so  $|D_{\sigma_1}| = 0$ ). We then split into two subcases based on the size of  $D_\alpha$  compared to the size of the entire bucket as we did in Case 1 and the proof proceeds in exactly the same manner.

□

## Optimality of Mastermind for $(n, k) = (4, 6)$

Up to isomorphism, the first guess for this game will be one of the following:

$$\{(0, 0, 0, 0), (0, 0, 0, 1), (0, 0, 1, 1), (0, 0, 1, 2), (0, 1, 2, 3)\}$$

Their respective worst-case responses and remaining solutions after that response are

$$\{(0, 0) \rightarrow 625, (1, 0) \rightarrow 317, (0, 0) \rightarrow 256, (0, 1) \rightarrow 276, (0, 2) \rightarrow 312\}$$

For each of these guesses, there exists a response that leaves at least 256 possible remaining solutions, as shown above. After this point, for every possible guess made, there will be at most fourteen possible responses given. Thus, at every step, the number of possible solutions remaining is reduced by at most a factor of fourteen. We use this information-theoretic lower bound to show that after the third guess, there will still be at least two solutions remaining:

$$\left\lceil \frac{\left\lceil \frac{256}{14} \right\rceil}{14} \right\rceil = 2$$

Thus no strategy could guess the hidden vector on the fourth turn, so it is not possible for a deterministic guessing strategy to guarantee finding the hidden vector in fewer than five turns.

## Lower Bound for Non-Adaptive Strategies: $n = k$

We consider the game of Mastermind for  $n = k$  with no repeats. To put a lower bound on non-adaptive strategies, we make the following observation: In order to distinguish between all possible solutions, every solution needs to have a distinct set of responses for the guesses made by a given strategy. Therefore, we will analyze all solutions for which every response is less than some number  $x$ . Let  $t$  be the number of guesses made. There will therefore be at most  $x^t$  possible response sets. The number of solutions which do not correspond to the responses for a single query is at most the sum of the sizes of the smaller buckets, which we define as buckets  $B_x$  through  $B_n$ . Thus after  $t$  guesses, the number of remaining solutions corresponding to these responses is at least

$$n! - t \sum_{i=x}^n \binom{n}{i} D(n-i)$$

As proven before, the above sum is at most  $n!/x!$ . Thus, the number of solutions is at least  $n!(1 - t/x!)$ . Now we let

$$t = \frac{n \log n}{(\log \log n)^{1+\epsilon}} \quad \text{and} \quad x = \log n$$

Thus we have

$$\begin{aligned} \log x^t &= \frac{n \log n}{(\log \log n)^{1+\epsilon}} \cdot \log \log n \\ &= \frac{n \log n}{(\log \log n)^\epsilon} \end{aligned}$$

For sufficiently large  $n$ ,

$$\frac{n \log n}{(\log \log n)^\epsilon} < \frac{n \log n}{4} < \frac{n(\log n - 1)}{2}$$

Therefore,

$$\begin{aligned} \log(x^t) &< \frac{1}{2}(n \log n - n) \\ &< \frac{1}{2} \log(n!) \quad \text{cf. Stirling's Approx.} \end{aligned}$$

Raising  $e$  to the power of both sides, we get

$$x^t < \sqrt{n!}$$

Similarly, we can bound  $\log x!$  below by

$$\begin{aligned} \log(x!) &> x \log x - x \\ &= \log n \cdot \log \log n - \log n \\ &= \log n(\log \log n - 1) \\ &> 4 \log n \quad \text{for sufficiently large } n \end{aligned}$$

Also

$$\begin{aligned}\log t &= \log n + \log \log n - \log((\log \log n)^{1+\epsilon}) \\ &< \log n + \log \log n \\ &< 2 \log n\end{aligned}$$

By exponentiating both sides, we get

$$x! > t^2$$

By combining these two results, we get

$$\frac{n!}{x^t} \left(1 - \frac{t}{x!}\right) \geq \sqrt{n!} \left(1 - \frac{1}{t}\right)$$

And for sufficiently large  $n$ ,

$$\sqrt{n!} \left(1 - \frac{1}{t}\right) > 1$$

Which gives the inequality

$$n! \left(1 - \frac{t}{x!}\right) > x^t$$

Therefore, for any  $\epsilon > 0$  there exists a number  $N$  such that for any  $n > N$ , for any set of

$$\frac{n \log n}{(\log \log n)^{1+\epsilon}}$$

guesses, there will be more solutions that have received only responses between 0 and  $x - 1$  than there will be response sets of those responses. Therefore it is impossible for any non-adaptive strategy to guarantee identifying the solution vector in fewer than the following number of guesses:

$$\frac{n \log n}{(\log \log n)^{1+\epsilon}}$$



## Lower Bound for Non-Adaptive Strategies Using Entropy

Our proof follows along similar lines as [Doerr et al]. Let  $q_i$  be the  $i$ -th guess of our deterministic strategy. Let  $s$  be the smallest index such that all  $n!$  codes in  $S_n$  are uniquely determined by the responses to  $q_1, q_2, \dots, q_s$ . Consider a code  $Z$  sampled uniformly at random from the set of permutations  $S_n$ . Then consider the random variables  $Y_i = \text{eq}(Z, q_i)$  be the response (# black hits) to guess  $q_i$ . Then the vector  $Y = (Y_1, Y_2, \dots, Y_s)$  always uniquely determines, and is uniquely determined by,  $Z$ . So  $H(Z) = H(Y)$ . Since  $Z$  is a random variable with  $n!$  outcomes of equal probability, we know  $H(Z) = \log_2 n!$ .

We know for any series of events, the entropy of the series as a whole is less than or equal to the sum of the individual entropies of the events. Using this, we get:

$$H(Y) \leq \sum_{i=1}^s H(Y_i).$$

So now we bound  $H(Y_i)$ . By its definition,

$$H(Y_i) = - \sum_{x=0}^n \Pr(Y_i = x) \cdot \log_2(\Pr[Y_i = x]).$$

What is  $\Pr[Y_i = x]$ ? This is the number of solution vectors that have  $x$  fixed points with respect to the query  $q_i$ . Using our previous terminology, this is bucket  $x$ , which, as we've shown before has size  $|B_x| \leq n!/x!$  (maybe footnote here). So we get  $\Pr(Y_i = x) \leq (n!/x!)/n! = 1/x!$ . So

$$\begin{aligned} H(Y_i) &\leq - \sum_{x=0}^n \frac{1}{x!} \cdot \log_2\left(\frac{1}{x!}\right) \\ &= \sum_{x=2}^n \frac{\log_2(x!)}{x!} \quad \text{zero for } x \in \{1, 2\} \\ &\leq \sum_{x=2}^n \frac{x \log_2 x}{x!} \\ &\leq \sum_{x=2}^n \frac{x(x-1)}{x!} \\ &= \sum_{x=2}^n \frac{1}{(x-2)!} \\ &\leq e \end{aligned}$$

Combining this with the above inequality gives  $H(Y) \leq s \cdot e$ . So we have  $\log_2(n!) \leq s \cdot e$  so  $s \geq \log_2(n!)/e$ .

This gives us a lower bound of  $O(n \log n)$  turns for any non-adaptive strategy for permutation mastermind.