Das Schachbrett-Rätsel

In dieser Lernaufgabe geht es um einen neuen, verblüffenden Kartentrick – das Schachbrett-Rätsel. Dieses Rätsel ermöglicht eine anschauliche Einführung einer effizienten 1-fehlerkorrigierenden Kodierung - der Hamming-Kodierung.

Ein neuer Kartentrick

Zwei Personen, ein Magier und ein Helfer, führen einen neuen Kartentrick vor. Auf ihren Karten steht auf einer Seite eine 1 und auf der anderen eine 0. Man kann diesen Trick aber auch mit beliebigen Karten spielen. Eine aufgedeckte Karte bedeutet 1 und eine verdeckte 0. Der Magier verlässt den Raum. Nun wählt ein Freiwilliger ein Feld eines Schachbretts aus, unter diesem Feld wird ein Schlüssel versteckt. Wir machen dies so, dass ein Papier mit Schlüsselsymbol auf diesem Feld platziert wird und dann sämtliche Felder des Schachbrettes mit den Karten zugedeckt werden, der Schlüssel ist also nicht mehr ersichtlich. Der Freiwillige kann nun zudem ein vollständig zufälliges Muster aus Nullen und Einsen legen.

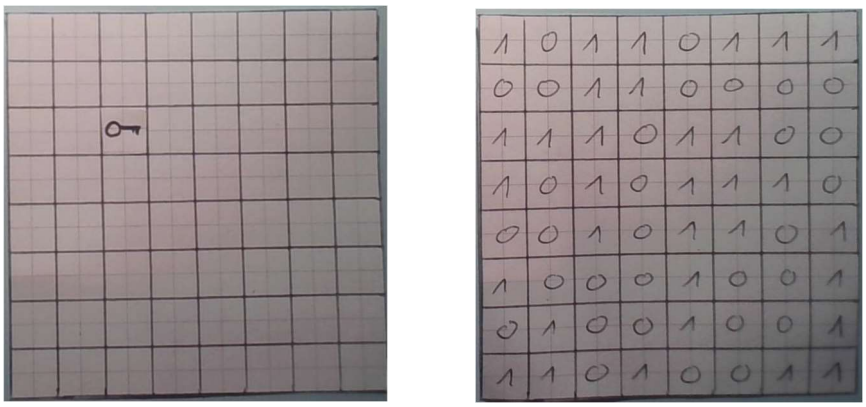


Bild 1: Schachbrett mit Schlüssel und Muster aus Einsen und Nullen

Da der Helfer des Magiers im Raum geblieben ist, kennt er das magische Quadrat, unter welchem sich der Schlüssel befindet. Dieser Helfer darf nun nur eine Kleinigkeit ändern, er darf genau eine Karte drehen.
Nun bittet man den Magier in den Raum und lässt ihn das magische Quadrat mit dem versteckten Schlüssel bestimmen.
Der Helfer im Raum hat die Karte, welche er dreht, so ausgewählt, dass der Magier das gesuchte magische Quadrat identifizieren kann. Wie genau sind die beiden, der Magier und der Helfer, vorgegangen?

Wie funktioniert der Trick?

Betrachten wir hier zunächst als Vereinfachung nur 4x4-Felder. Die Ausgangslage ist: Der Freiwillige setzt den Schlüssel (linkes Bild) und wählt ein Muster aus Nullen und Einsen (rechtes Bild).

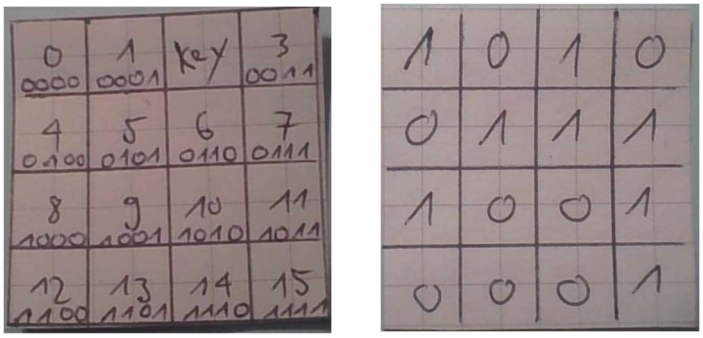


Bild 2: Dies ist die Ausgangslage mit Schlüssel und zufälligem Muster aus Nullen und Einsen; die 16 Felder nummerieren wir zeilenweise binär von Null bis 15, somit erhält jedes Feld eine eindeutige Adresse.

Die Idee ist, geschickte Teilmengen dieser 4x4-Felder zu wählen, und Paritäten (modulare Summen der Feldinhalte dieser Gebiete) zu bestimmen; der Helfer wird die 4 Werte dazu verwenden, dem Magier die Adresse des Feldes mit dem Schloss mitzuteilen.

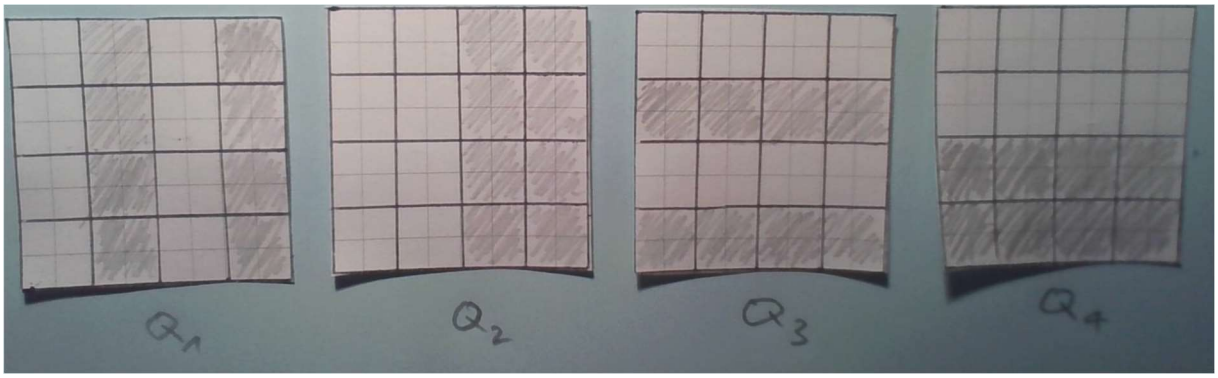


Bild 3: Das sind die vier Teilmengen Q1 – Q4, in welchen wir Paritäten bestimmen.

Wenn wir im obigen Beispiel die Paritäten, also Summen mod 2, bestimmen, erhalten wir in Q1 eine 0, in Q2 eine 1, in Q3 eine 0 und in Q4 eine 1. Das Ziel wäre jetzt eine Karte so zu flippen, dass danach mit den Paritäten in Q1 bis Q4 das Feld des Schlüssels, also 0010, kommuniziert werden kann. Schematisch dargestellt haben wir also folgende Situation:

Aktuell:	1010	
Wir müssen ändern:	1000	(weil xor bzw. mod 2, ergibt 1010 + 1000 = 0010)
Ziel:	0010	

Wenn der Helfer die Paritäten von Q3, Q2, Q1, Q0 ausrechnet und ihre Werte in dieser Reihenfolge nicht die Adresse des Feldes mit dem Schlüssel beschreiben, dann will er die Paritäten der von der Adresse abweichenden Qs ändern bzw. flippen. Das erreicht er, indem er die Karte umflippt, welche genau in diesen Gebieten und nicht in den Gebieten mit richtigen Werten liegt.

Aufgabe 1:	Suchen Sie eine Karte, die a) keinen Wert von einem Q ändert, b) die Werte von Q1 und Q4 ändert, aber Q2 und Q3 unverändert lässt, c) den Wert von Q2 ändert und keinen anderen.

Wenn der Helfer jetzt also das Feld mit der Binärnummer 1000 flippt, ergibt sich folgendes Muster:

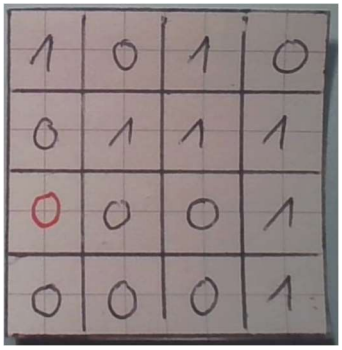


Bild 4: Muster mit geflipptem Feld (rot)

In diesem Muster ergeben die Paritäten in Q1 bis Q4: In Q1 eine 0, in Q2 eine 1, in Q3 eine 0 und in Q4 eine 0, also 0010, den Code für das magische Quadrat mit dem Schlüssel.

Für das Schachbrett (8x8-Felder) funktioniert der Trick ganz analog.

Warum funktioniert das?

Auch für diese Erklärung beschränken wir uns der Einfachheit halber auf 4x4-Felder.

Statt die vier Paritäten in Q1 bis Q4 zu bestimmen, können die Binärcodes der Felder mit einer Eins addiert werden (mod 2 bzw. xor). Der Binärcode eines Feldes widerspiegelt nämlich genau, zu welchen Teilmengen Q1 bis Q4 das Feld gehört:

0000 bedeutet ‘nicht in Q1‘, ‘nicht in Q2‘, …;
1011 bedeutet ‘in Q1‘, ‘in Q2‘, ‘nicht in Q3‘, ‘in Q4‘.

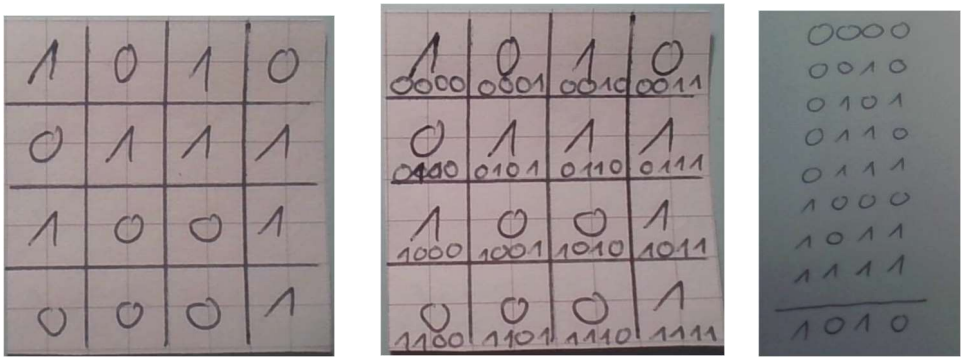


Bild 5: Ausgangslage (links), binär nummerierte Felder (Mitte), Summe aller auf Eins geflippten Binärcodes (rechts)

Aus der aktuellen Summe kann immer mit einem Flip die gewünschte Zielsumme erreicht werden. Dazu muss die Differenz aufaddiert werden. Weil Addition und Subtraktion (mod 2) gleichbedeutend sind und alle 4stelligen Binärcodes in den 16 Feldern vorkommen, bringt ein Flip (in unserem Fall von Feld 1000) das gewünschte Ergebnis.

Bemerkung: Die Teilmengen Q1 bis Q4 können statt grafisch auch wie folgt beschrieben werden:

Q1: Ist die Binärziffer des Feldes von der Form _ _ _ 1 ?
Q2: Ist die Binärziffer des Feldes von der Form _ _ 1 _ ?
Q3: Ist die Binärziffer des Feldes von der Form _ 1 _ _ ?
Q4: Ist die Binärziffer des Feldes von der Form 1 _ _ _ ?

Aufgabe 2:	Besprechen Sie zu zweit den Trick und die Rollen von Helfer und Magier für 8x8-Felder. Sie können dazu die in Bild 1 abgebildete Ausgangssituation annehmen. Erklären Sie insbesondere, für welche Teilmengen Q1, …, Qn die Paritäten bestimmt werden müssen. Sie sollten den Trick am Ende für 4x4- bzw. 8x8-Felder vor Publikum vorführen können.

Wie hängt das Rätsel mit dem Thema „selbstkorrigierende Kodierungen“ zusammen?

Wir betrachten einen Codeblock von 4x4 Bits:

Um eine 1-fehlerkorrigierende Codierung zu erhalten, verwenden wir nicht alle 16 Bits als Nachrichtenbit, sondern 4 geeignete Kontrollbits. Wir verwenden nämlich für jede Teilmenge Q1 bis Q4 ein Paritätsbit und setzen dieses in die Felder mit Binärziffern 0001, 0010, 0100 und 1000. Kurz: Wir wählen als Kontrollbits diejenigen 4 Felder, die genau in einem der vier Gebiete Q1 bis Q4 liegen. Dann können 11 Nachrichtenbits gesetzt werden, das Bit an der Stelle 0 bleibt vorerst leer, auch dieses Bit dient der Fehlerkorrektur.

Aufgabe 3:	Hat diese Wahl etwas Gemeinsames mit der Wahl der Kontrollbits in der Unterrichtseinheit (Lehrmittel) zur 1-fehlerkorrigierenden Kodierung mit minimaler Anzahl von Kontrollbits?

Beispiel: 0110001110 sind die Nachrichtenbits; diese werden z.B. von links beginnend in aufsteigende Felder eingefügt, danach werden die vier Paritätsbits gesetzt. Zuletzt wird das Bit auf Feld 0000 als Paritätsbit des ganzen Codeblocks gesetzt.

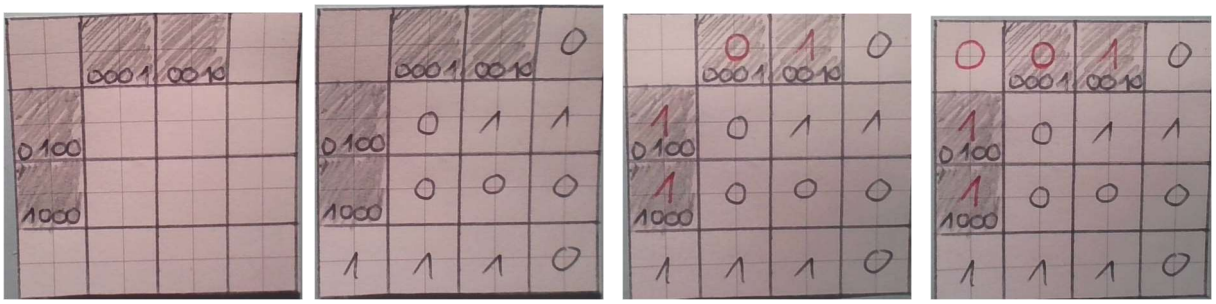
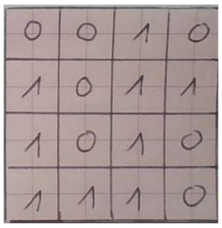


Bild 6: 4x4-Codeblock mit für Paritätsbits reservierten Plätzen (links), mit den Nachrichtenbits (Mitte links), mit den ersten vier Paritätsbits für Q1 bis Q4 (Mitte rechts), mit dem 5. Kontrollbit (rechts)

Der Code ist damit 1-fehlerkorrigierend und zusätzlich wird 2-fehlererkennend, falls nämlich die ersten vier Kontrollbits keinen Fehler detektieren, aber das 5. Kontrollbit doch, dann liegen (mindestens) zwei Fehler vor.

Aufgabe 4:	Seien 11100010001 die elf Nachrichtenbits. Bilden Sie den 4x4-Codeblock mit den fünf zusätzlichen Kontrollbits.

Aufgabe 5:	Angenommen der Empfänger erhält folgenden beschädigten 4x4-Codeblock. Bestimmen Sie die ursprünglichen elf Nachrichtenbits (das gesendete Code-Wort), wenn Sie annehmen, dass genau 1 Bit umgeflippt wurde.



Die Anordnung der 4x4-Felder in einem Quadrat ist nur eine Darstellung. In einer Reihe angeordnet, wird ersichtlich, dass sich die Kontrollbits an 1., 2., 4., 8., 16. Stelle usw. - alles Zweierpotenzen - befinden. Für 2^m viele Nachrichten brauchen wir also neben den m Nachrichtenbits auch eine Anzahl $\lfloor \log_2 m \rfloor + 1$ Kontrollbits. Die Zahl $\lfloor \log_2 m \rfloor$ ist der diskrete Logarithmus von m bei der Basis 2, d.h. diejenige natürliche Zahl x, die $2^{x-1} \leq m \leq 2^x$ erfüllt. Diese effiziente Kodierung nennt man Hamming-Kodierung: Für 2^m viele Nachrichten reichen Code-Wörter der Länge $m + \lfloor \log_2 m \rfloor + 1$ aus, um die Korrektur eines Fehlers zu garantieren.

Bild 7: Dies ist eine Liste von Bits, wobei die notwendigen Kontrollbits hervorgehoben sind.

Die Effizienz der Hamming-Kodierung zeigt sich anschaulich bei grösseren Codeblöcken, für jede Verdoppelung des 4x4-Codeblocks wird nur ein Kontrollbit mehr benötigt.

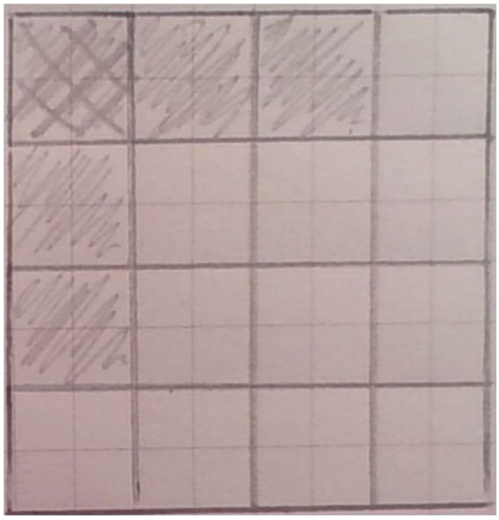
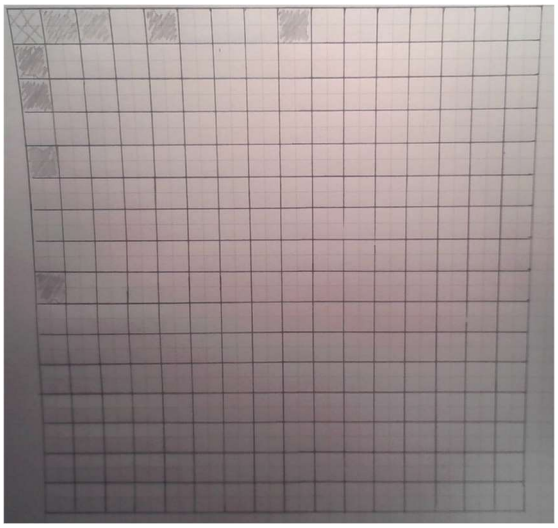



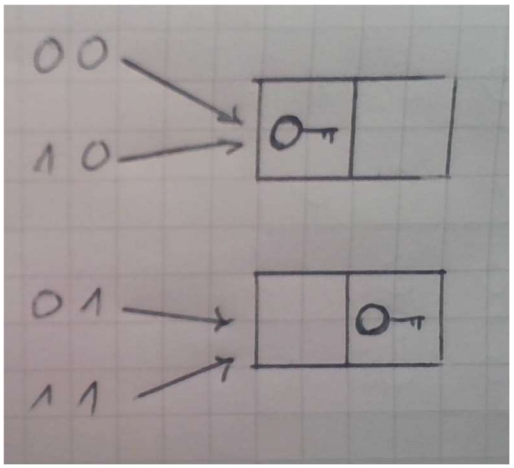
Bild 8: 4x4-Codeblock mit 5 Kontrollbits (31.25%), 16x16-Codeblock mit den 9 Kontrollbits (3.5%)

Zusammenfassung:

Wir haben einen weiteren, verblüffenden Kartentrick gesehen und über das Verständnis dieses Kartentricks konnten wir auf anschauliche Art und Weise den Bau einer optimalen 1-fehlerkorrigierenden Hamming-Kodierung nachvollziehen.

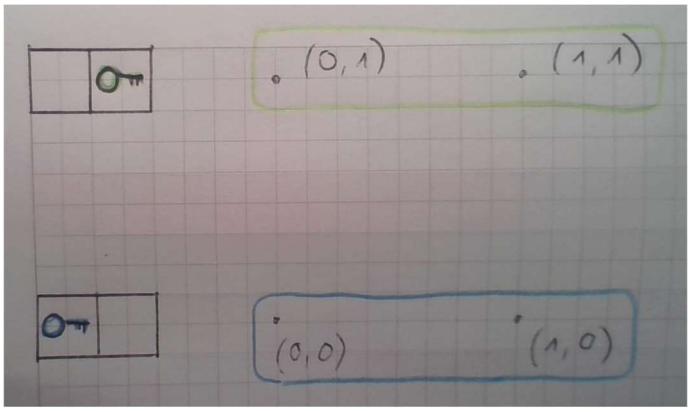
Ergänzung: Warum funktioniert der Trick nicht allgemein für n Felder, aber für 4x4-Felder bzw. 8x8-Felder?

Um einen guten Einblick zu geben, vereinfachen wir die Situation und betrachten zunächst nur zwei Felder statt 8x8-Felder. Der Schlüssel wird also in einem von zwei möglichen Feldern versteckt. Nun werden die beiden Felder mit Karten verdeckt und es wird ein zufälliges Muster aus Nullen und Einsen gelegt. Die Magier können vereinbaren, dass die zweite Karte anzeigt, ob der Schlüssel darunter verborgen ist: Zeigt die zweite Karte eine 1, ist der Schlüssel darunter; zeigt die zweite Karte eine 0, ist der Schlüssel im ersten Feld zu finden. Der Magier im Raum, welcher die Position des Schlüssels kennt, kann immer einen Flip vornehmen, so dass die zweite Karte danach diese Information über die Position des magischen Quadrates trägt. Falls die zweite Karte von Anfang an die richtige Information trägt, flippt der Magier die erste Karte, sonst die zweite Karte.

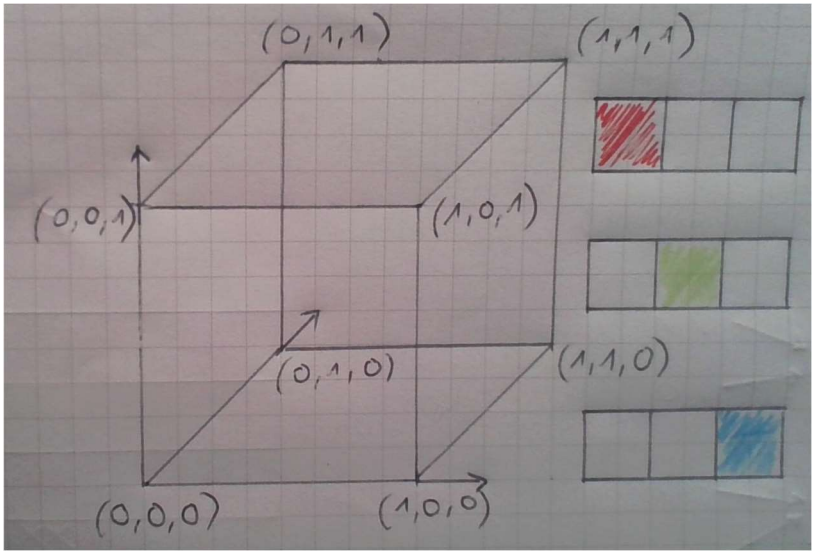


Wir können die Situation mit vier Punkten und ihren Koordinaten (0,0), (1,0), (0,1) und (1,1) illustrieren. Der Flip einer Karte entspricht dann einer Bewegung von einer Ecke zu einer benachbarten Ecke. Mit der

obigen Abmachung, dass die zweite Karte anzeigen soll, wo sich der Schlüssel befindet, würden also (0,0) und (1,0) den Schlüssel im Feld 0 signalisieren, umgekehrt (0,1) und (1,1) weisen den Schlüssel dem Feld 1 zu. Der Wert K, mit welchem der Magier das magische Quadrat bestimmt, wird also wie folgt berechnet: K = (0*a + 1*b) mod 2, wobei a dem Wert der Karte auf Feld 0 und b dem Wert der Karte auf Feld 1 entspricht.



Bei drei Karten taucht ein Problem auf. Wenn wir die geometrische Interpretation der Karten betrachten, erhalten wir die Koordinaten eines Würfels. Bei jedem Flip bewegt man sich entlang einer Kante des Würfels. Eine Strategie für das Rätsel zu haben, würde bedeuten, dass der 2. Magier, wenn er in den Raum kommt, jedem Zustand der drei Bits eindeutig das magische Quadrat zuweisen könnte. Wir können dies mit drei Farben visualisieren, rot für Feld 0, grün für Feld 1 und blau für Feld 2. Eine Strategie zu haben entspricht also einer Einfärbung der acht Ecken: Werden alle Ecken rot gefärbt, tippt der 2. Magier immer auf Feld 0.



Eine andere Strategie wäre, die Summe der Werte auf den Feldern 0 und 1 zu bilden. Wie viele Strategien gibt es? Es gibt 3^8 Strategien. Für den 8x8-Würfel gäbe es dann $64^{(2^{64})}$ Strategien. Eine andere Strategie könnte sein, den Wert, mit welchem der Magier das magische Quadrat bestimmt, wie folgt zu berechnen: K = (0*a + 1*b + 2*c) mod 3, wobei a der Wert der Karte in Feld 0 ist, usw. Entsprechend könnte dann für das 8*8-Feld der Wert K berechnet werden: K $=(0*x_0+1*x_1+2*x_3+\ldots+63*x_{63})$ mod 64. Dies wäre eine andere Strategie, die Ecken unseres Würfels einzufärben. Im Fall von (0,0,0) können wir durch Ändern eines Bits alle drei möglichen Zustände erreichen. Im Fall von (0,1,0) aber können wir durch Ändern eines Bits nicht den Wert 2 erhalten. Diese Strategie würde also für unsere beiden Magier nicht taugen. Vielleicht gibt es aber eine Strategie, gibt es eine Färbung, so dass die Nachbarecken jeder Ecke rot, blau und grün sind? Das würde für unser

Puzzle bedeuten, dass der 1. Magier bei jeder Konstellation der Bits mit einem Flip die Schlüsselposition kommunizieren könnte. Unsere Aufgabe lautet: Finde eine passende Färbung oder zeige, dass es keine passende Färbung gibt. Eine Möglichkeit ist, alle Ecken durchzugehen und zu zählen, wie viele Nachbarecken eine spezifische Farbe haben, z.B. rot. Jede Ecke soll genau eine rote Nachbarecke haben, daher müsste man auf ein Total von 8 roten Nachbarecken kommen. Andererseits wird aber jede rote Ecke genau dreifach gezählt, nämlich für jede Ecke, an welche sie grenzt, daher müsste das Total von roten Nachbarecken ein Vielfaches von 3 sein. Widerspruch! Dies lässt sich auf höhere Dimensionen übertragen, auch wenn wir uns die dazugehörigen n-dimensionalen Würfel nicht mehr vorstellen können: In einer Ecke gibt es immer n verschiedene Nachbarn, insgesamt gibt es 2^n Ecken, wie im Dreidimensionalen können wir wieder die totale Anzahl roter Nachbarecken zählen. Einerseits muss dies 2^n ergeben, eine für jede Ecke, andererseits wird jede rote Ecke für jeden ihrer Nachbarn gezählt, also $n \cdot (\#\ rote\ Ecken)$. Also gilt: $2^n = n \cdot (\#\ rote\ Ecken)$. Weil die linke Seite eine Zweierpotenz ist, muss die rechte Seite ebenfalls eine Zweierpotenz sein, was nur geht, falls n selbst eine kleinere Zweierpotenz ist, z.B. n=2, n =4 oder n=8.