.

Selbstkorrigierende Kodierungen

Erweiterung auf k-fehlerkorrigierende Kodierungen

HS 2024
25.01.2025

EINGEREICHT BEI
ETH Zürich
D-INFK
Fachdidaktik Informatik
Prof. Dr. Juraj Hromkovic

EINGEREICHT VON
Jonas Binz, Simon Häberli & Daniel Sparber

Inhaltsverzeichnis

1. Konzeption der Unterrichtseinheit

1.1. Analyse des Vorwissens der Klasse

Selbstkorrigierende Kodierungen

Die Klasse hat ein Grundwissen zu selbstkorrigierenden Kodierungen, welches zum Beispiel im Kapitel 4 des Lehrmittels „Informatik - Data Science und Sicherheit“ vermittelt wird. Insbesondere beinhaltet dies die Bekannschaft mit folgenden Begriffen:

- Symbole
- Prüfsymbol
- Kodierung
- Code-Wörter
- Länge einer Kodierung
- k-fehlererkennend
- k-fehlerkorrigierend
- Hamming Abstand
- Abstand einer Kodierung

Die Schüler:innen können den Unterschied zwischen k-fehlererkennenden und k -fehlerkorrigierenden Kodierungen erklären und kennen Beispiele von 1-fehlererkennenden Kodierungen, welche nicht 1-fehlerkorrigierend sind. Sie berechnen die Länge von Kodierungen in einigen Beispielen korrekt und vergleichen diese mit der Anzahl dabei kodierter Nachrichten.

Zahlensysteme

- Binärzahlen
- Umwandlung vom Dezimalsystem ins Binärsystem und umgekehrt.

Spezifisch für den Kartentrick

- Die Schüler:innen können die 1-fehlerkorrigierende Kodierung des Kartentricks in Beispielen durchführen.
- Sie erklären mühelos, wieso diese Kodierung tatsächlich 1-fehlerkorrigierend ist.
- Die Schüer:innen wissen, dass der kennengelernte Kartentrick für 2^m Nachrichten eine Kodierung von ungefähr der Länge $m + 2\sqrt{m}$ bildet, und können dies begründen.
- Die Addition modulo 2 ist den Schüler:innen vertraut und sie verstehen die Anwendung davon im Kartentrick.

Spezifisch für das Reed-Solomon Verfahren

Das Reed-Solomon Verfahren basiert auf linearen Polynome. Ein Grundwissen in diesem Thema ist somit massgebend. Insbesondere:

- Bestimmung der Steigung und des Achsenabschnitts der zu einem linearen Polynom gehörigen Geraden.
- Eindeutige Bestimmung einer Geraden durch zwei Punkte.
- Graphische Darstellung von Punkten und Geraden in der Koordinatenebene

1.2. Leitidee

Selbstkorrigierende Kodierungen sind allgegenwärtig in unserem heutigen Leben. Sie erlauben uns unter schwierigen Bedingungen Daten sicher von A nach B zu schicken. Falls die Übertragung beeinträchtigt wird, können diese Daten automatisch wiederhergestellt werden.

Solche Kodierungen zu verstehen und anwenden zu können gibt den Schüler:innen ein solides Grundverständnis in der Informationstechnologie, fördert das algorithmische Denken, und bereitet sie auf ein weiterführendes Studium vor.

1.3. Dispositionsziele

1. Die Schüler:innen lernen bekannte Methoden zu erweitern, um somit selbst neue fehlerkorrigierende Kodierung analysieren und entwerfen zu können.
2. Die Schüler:innen sind in der Lage mithilfe von grafischen Darstellungen Kodierungsfehler zu entdecken.
3. Die Schüler:innen erweitern ihr abstraktes Denken gegenüber von linearen Gleichungen, sie sehen den Zusammenhang zwischen Funktionen und Kodierungen.
4. Die Schüler:innen lernen, wie man verschiedene Algorithmen miteinander vergleicht und somit Vor- und Nachteile der Algorithmen erkennt.
5. Die Schüler:innen werden sich bewusst wie wichtig die korrekte Datenübermittlung für viele alltägliche Bereiche ist, und sind motiviert das Thema selbständig weiter zu verfolgen.

1.4. Operationaliserte Lernziele

1. Der Schüler legt bei einem Kartentrickspiel mit $a \cdot b$ Karten auf Anhieb $2(a + b + 2)$ korrekte Kontrollkarten für den Fall von 2 geänderten Karten.
2. Eine Schülerin erklärt einem Mitschüler mittels Beispiele, wieso der oben angewandte Kartentrick zur Korrektur von bis zu zwei Fehlern funktioniert.
3. Die Schüler:innen kodieren vorgegebene Karten mit Kontrollkarten auf eine 3-fehlerkorrigierende Weise.
4. Die Schüler:innen erkennen aus der graphischen Darstellung von Punkten, welche mit dem Reed-Solomon-Verfahren übermittelt wurden, ob bei der Uebermittlung Fehler passierten, und können diese, falls diese Fehler bestimmbar sind, korrigieren.
5. Die Schüler:innen erklären durch Angabe eines Gegenbeispiels, wieso $2k + 1$ übermittelte Punkte beim Reed-Solomon-Verfahren nicht genügen, um k Fehler zu korrigieren.
6. Die Schüler:innen nennen die Mindestanzahl benötigter Punkte für ein k-fehlerkorrigierendes Reed-Solomon-Verfahren und erklären dies, zum Beispiel durch Kontrastierung mit dem vorderen Gegenbeispiel.

2. Leitprogrammartige Unterrichtsunterlagen

2.1. Erweiterter Kartentrick

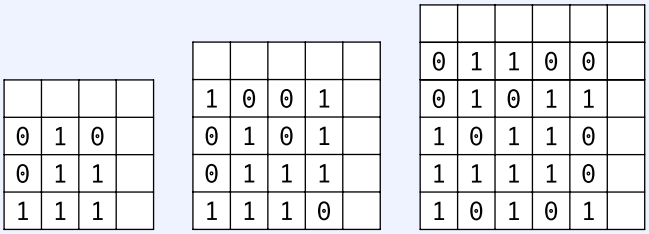
2.1.1. Auffrischen des Kartentricks

In den vorhergehenden Lektionen haben Sie gesehen, wie mithilfe des Kartentricks einfach Kontrollbits berechnet werden können und mithilfe dieser Kontrollbits Fehler erkannt und korrigiert werden können.

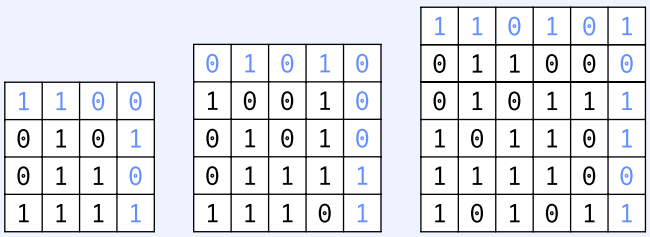
Zur Erinnerung: beim Kartentrick wird eine Matrix um eine Zeile und eine Spalte von Kontrollbits ergänzt. Die Kontrollbits werden so gewählt, dass für jede Zeile bzw. Spalte die Summe aller Einsen gerade ist.

Aufgabe

Ergänzen Sie folgendes Kartenspiel mit Kontrollkarten, welche einen Fehler erkennen und korrigieren können.



Lösung



Wenn ein Fehler aufgetreten ist, stimmen nicht mehr alle Kontrollbits. Die Anzahl Einsen in der Zeile des Fehlers ist nun nicht mehr gerade. Dasselbe gilt für die Spalte des Fehlers. Deshalb können wir einen unbekannten Fehler dort lokalisieren, wo sich diese Zeile und Spalte kreuzen.

Aufgabe

In folgendem Kartenspiel mit Kontrollkarten wurde eine Karte umgedreht. Zeichnen Sie die umgedrehte Karte ein. Markieren Sie zudem die Kontrollkarten, welche die umgedrehte Karte anzeigen, sowie ihre entsprechende Zeile, bzw. Spalte.

1	0	0	1	1	1
0	0	1	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	0	1	0	0
1	1	0	0	1	1

Lösung

1	0	0	1	1	1
0	0	1	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	0	1	0	0
1	1	0	0	1	1

2.1.2. Zwei Fehler korrigieren

In vielen Fällen wollen wir mehr als nur einen Fehler erkennen und korrigieren können. Dafür konstruieren wir gemeinsam eine Erweiterung des Kartentricks. Im folgenden Abschnitt sehen wir uns an, was nötig ist um bis zu zwei Fehler zu erkennen.

2.1.2.1. Martins Behauptung

Martin hat den Kartentrick bereits gut einstudiert und mehrfach vorgeführt. Er hat dabei gemerkt, dass man auch erkennt, wenn zwei oder mehr Karten umgedreht werden. Er behaupted deswegen, dass gar keine Anpassung nötig ist und der Kartentrick direkt auch für zwei Fehler anwendbar ist.

Aufgabe

Im folgenden sind ein paar Beispiele die die Kartentrickcodierung verwenden. Leider haben sich bei den Nachrichten je zwei Fehler eingeschlichen. Versuchen Sie bei allen Nachrichten beide Fehler zu finden.

1	1	0	1	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	0	1	1	0	0
1	0	1	0	1	1

0	1	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	0

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

Lösung

Matrix 1: In der folgenden Abbildung sind alle Zeilen und Spalten markiert, deren Anzahl an Einsen ungerade ist. Wir haben je zwei solche Zeilen und Spalten.

1	1	0	1	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	0	1	1	0	0
1	0	1	0	1	1

Beide der folgenden Fehlerpositionen sind möglich:

1	1	0	1	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	0	1	1	0	0
1	0	1	0	1	1

1	1	0	1	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	0	1	1	0	0
1	0	1	0	1	1

Wir können somit leider nicht genau feststellen wo die Fehler aufgetreten sind, und können die Fehler folglich nicht korrigieren.

Matrix 2: Alle Spalten haben eine gerade Anzahl an Einsen. Zwei Zeilen haben eine ungerade Anzahl. Diese sind in der folgenden Abbildung hervorgehoben:

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

Auch hier können wir nicht genau sagen, wo die Fehler aufgetreten sind. Die Fehler müssen beide auf der selben Spalte liegen, da sonst nicht alle Kontrollbits der Spalten korrekt wären. Wir haben in Summe sechs Spalten und somit sechs mögliche Fehleranordnungen die zu dieser Matrix geführt haben können. Im Folgenden sind ein paar Möglichkeiten aufgeführt:

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

...

Matrix 3: In der folgenden Abbildung sind alle Zeilen und Spalten markiert, deren Anzahl an Einsen ungerade ist.

0	1	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	0

Auch hier können wir nicht genau sagen, wo die Fehler aufgetreten sind. Beide der folgenden Fehlerpositionen sind möglich:

0	1	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	0

0	1	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	0

2.1.2.2. Ellas Idee

Ella möchte zusätzliche Kontrollbits einzuführen, um auch mehr als einen Fehler korrigieren zu können. Hierfür nimmt sie an, dass das Kartenrechteck mitsamt der bereits gelegten Kontrollkarten eine gerade Seitenlängen hat. Sie schlägt vor, das Rechteck in 2x2 Blöcke einzuteilen und für jeden Block ein zusätzliches Kontrollbit einzuführen. Das Kontrollbit eines Blocks wählt sie jeweils so, dass die Summe der Blockeinträge plus das Kontrollbit gerade ist.

1	1	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

1	0	1
0	1	1
1	1	0

Aufgabe

Führen Sie den Kartentrick für folgende Nachrichten aus. Fügen Sie zuerst die aus den vorhergehenden Lektionen bekannten Kontrollbits ein. Berechnen Sie dann die zusätzlichen Kontrollbits für folgende Nachrichten:

(a) `110110101`
(b) `111001111110000`
(c) `100100110101001111001110`

Lösung

(a)

1	0	1	0
1	1	0	0
1	1	0	0

1	1
1	1

1	0	1	0

(b)

1	0	0	1	1	1
1	1	1	0	0	1
1	1	1	1	1	1
1	0	0	0	0	1

1	0	1
1	0	1

(c)

0	0	1	1	0	0
1	0	0	1	0	0
0	1	1	0	1	1
0	1	0	0	1	0
1	1	1	1	0	0
0	1	1	1	0	1

1	1	0
0	1	1
1	0	1

Aufgabe

Lokalisieren Sie die zwei Fehler, die sich bei jeder der folgenden Nachrichten eingeschlichen haben. Um die Arbeit ein bisschen zu erleichtern, sind jene Zeilen und Spalten, die eine ungerade Anzahl Einsen haben bereits hervorgehoben.

1	1	0	1	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	0	1	1	0	0
1	0	1	0	1	1

1	0	1
0	1	1
1	1	0

0	1	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	0

1	0	1
0	1	1
1	1	0

Lösung

Wir können die Fehler mit Hilfe der zusätzlichen Kontrollbits in beiden Nachrichten exakt lokalisieren und somit auch korrigieren.

1	1	0	1	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	0	1	1	0	0
1	0	1	0	1	1

0	1	0	1	0	1
0	1	1	0	0	0
0	1	0	1	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	0

Sieht schon mal vielversprechend aus. Nun möchten wir herausfinden ob wir einfach Glück mit den Beispielen hatten, oder ob das Verfahren immer funktioniert.

Aufgabe

Probieren Sie Ellas Idee weiter aus. Versuchen sie entweder ein Gegenbeispiel zu finden, bei dem zwei Fehler nicht eindeutig erkennbar sind, oder erklären Sie kurz und bündig warum das Verfahren immer funktioniert.

Lösung

Im Allgemeinen funktioniert der Trick mit diesen zusätzlichen Kontrollbits **nicht**. Gegenbeispiele können auf mehrere Arten erstellt werden.

Zum Beispiel:

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

1	0	1
0	1	1
1	1	0

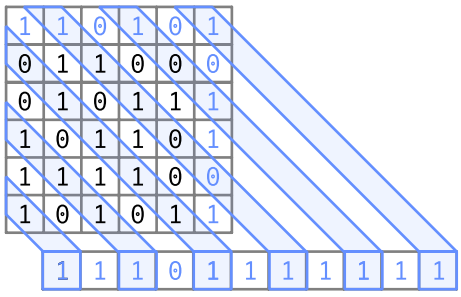
Bei diesem Gegenbeispiel ist die Idee, die zwei Fehler in die gleiche Spalte zu legen. Somit haben alle Spalten eine gerade Anzahl Einsen und die Spaltenkontrollbits geben keine Auskunft. Weiters können wir mit den zusätzlichen Kontrollbits den Fehler auf zwei Spalten einschränken. Das ist zwar eine Verbesserung, aber lässt immer noch folgende zwei Möglichkeiten zu.

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

1	1	0	0	0	1
0	1	1	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	1	1	0	0
1	0	1	0	1	1

2.1.2.3. Franziskas Idee

Franziska überlegt sich, zusätzliche Kontrollbits anhand von Diagonalen zu konstruieren. Sie geht dafür folgendermassen vor:



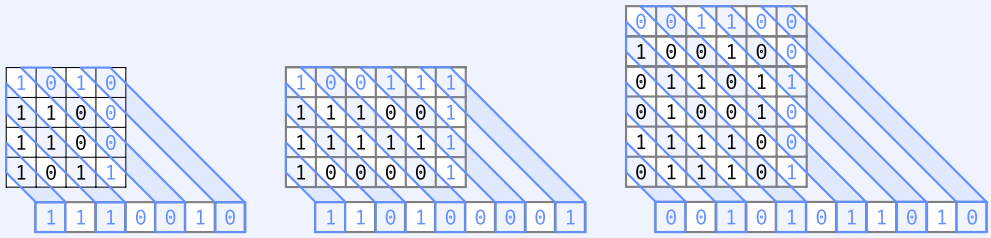
Die neuen Kontrollbits für die Diagonalen prüfen alle Datenbits, plus die bestehenden Kontrollbits aus dem einfachen Kartentrick. Somit wird alles dreifach überwacht.

Aufgabe

Führen Sie diese Idee für folgende Nachrichten aus, und berechnen Sie die zusätzlichen diagonalen Kontrollbits für folgende Nachrichten:

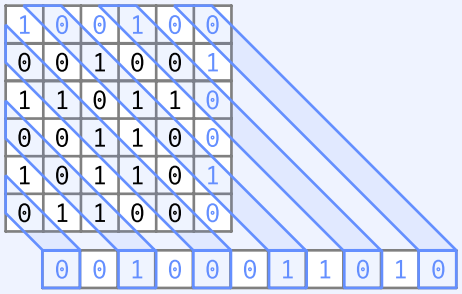
(a) `110110101`
(b) `111001111110000`
(c) `100100110101001111001110`

Lösung



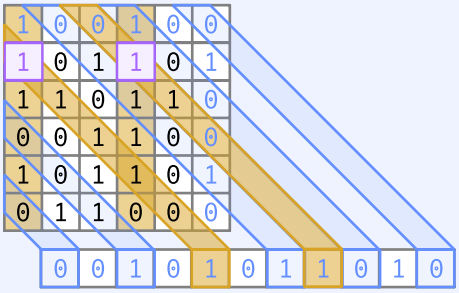
Aufgabe

Lokalisieren Sie alle Fehler, die sich bei der folgenden Nachricht eingeschlichen haben. Korrigieren Sie diese im Anschluss.



Lösung

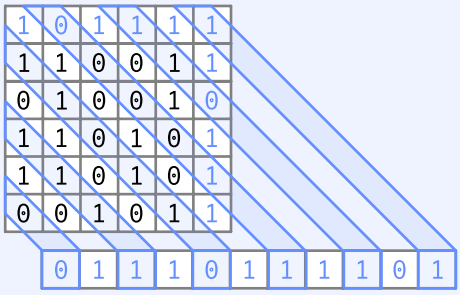
In der folgenden Abbildung ist die korrigierte Nachricht zu sehen. Die Korrekturen sind violett markiert. Wir finden die Fehler indem wir zuerst alle fehlerhaften Zeilen, Spalten und Diagonalen gelb markieren. Es gibt zwar keine fehlerhaften Zeilen, aber wir können die Fehler trotzdem finden, indem wir die Punkte an den sich die fehlerhafte Spalten mit den fehlerhaften Diagonalen schneiden markieren.

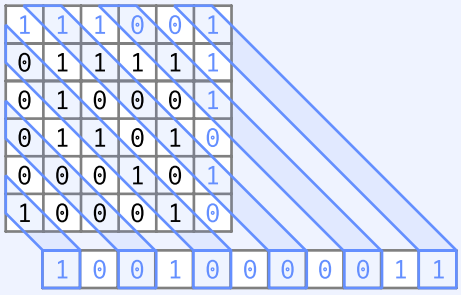


Franziska's Kodierung hat bei diesem Beispiel gut geklappt. Aber vielleicht war das auch nur Glück.

Aufgabe

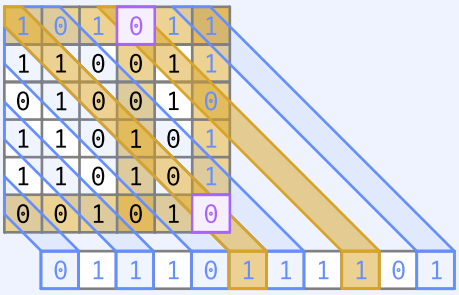
Versuchen Sie bei den folgenden zwei Nachrichten je beide Fehler zu finden und falls möglich zu korrigieren.





Lösung

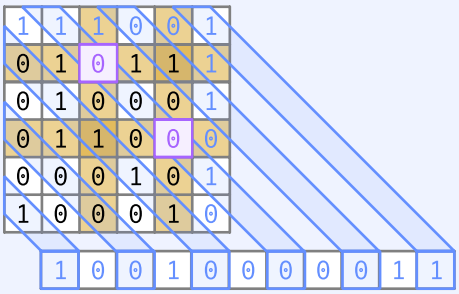
Nachricht 1: Zuerst markieren wir alle Spalten, Zeilen und Diagonalen bei denen die die Anzahl der Einsen ungerade ist. Wir haben wie in der folgenden Abbildung zu sehen je zwei Zeilen, Spalten und Diagonalen die Fehlerhaft sind. Wenn wir die beiden Bits korrigieren die an den zwei Punkten sind, wo sich je eine fehlerhafte Zeile, Spalte und Diagonale treffen, erhalten wir die korrigierte Nachricht:



Nachricht 2: Wir markieren wieder alle Zeilen, Spalten und Diagonalen die fehlerhaft sind. Dieses mal bekommen wir dabei zwei Spalten und zwei Zeilen, aber keine Diagonalen.

Wir haben vier Schnittpunkte und somit vier positionen wo die Fehler liegen können. Wir stellen fest, dass die Fehler weder auf der selben Spalte noch auf der selben Zeile liegen können, da sie sich sonst aufheben würden. Sprich, die Fehler müssen an zwei gegeüberliegenden Schnittpunkten liegen.

Würden die Fehler auf verschiedenen Diagonalen liegen, hätten wir auch zwei fehlerhafte Diagonalen, was aber nicht der Fall ist. Deswegen müssen die Fehler auf der selben Diagonale sein und sich gegenseitig aufheben. Wir bekommen folgende Lösung:



Nun, da wir gesehen haben, dass die Kodierung zu funktionieren scheint, würden wir gerne wissen wie effizient sie ist.. Wir können dafür die Anzahl Kontrollbits bei den vorhergehenden Übungsaufgaben zählen. Aber noch lieber würden wie die Effizienz für beliebig lange Nachrichten kennen.

Aufgabe

Angenommen die Karten sind quadratisch angeordnet und kodieren 2^m Nachrichten. Welche Länge hat die obige 2-fehlerkorrigierende Kodierung der 2^m Nachrichten ungefähr?

Lösung

Das Quadrat hat ungefähr Länge $\sqrt{m}$. Somit werden ungefähr $4\sqrt{m}$ Kontrollkarten gelegt. Die Länge der 2-fehlerkorrigierenden Kodierung ist also ungefähr $m + 4\sqrt{m}$.

Franziska behauptet ihre Lösung funktioniere immer. Sie möchte das mit den folgenden zwei Fällen verdeutlichen.

(I) Die zwei Fehler sind in verschiedenen Zeilen und Spalten.
(II) Die zwei Fehler sind in einer gemeinsamen Zeile oder Spalte.

Aufgabe

Angenommen zwei Fehler sind auf verschiedenen Zeilen und Spalten. Erklären Sie warum mit Hilfe der zusätzlichen diagonalen Kontrollbits beide Fehler genau lokalisiert werden können.

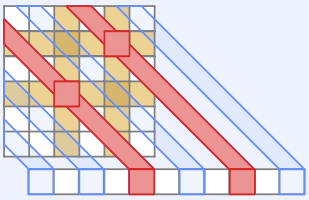
Lösung

Angenommen zwei Fehler liegen auf unterschiedlichen Spalten und Zeilen. In diesem Fall haben wir genau zwei Spalten plus zwei Zeilen bei denen die Anzahl der Einsen ungerade ist. Zwischen diesen Zeilen und Spalten gibt es genau vier Kreuzungspunkte, auf zwei von welchen die Fehler liegen müssen.

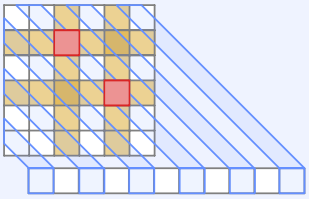
Weiters müssen die Fehler in den jeweils diagonal gegenüberliegenden Schnittpunkten sein, da sie sonst entgegen der Annahme nicht auf verschiedenen Zeilen bzw. Spalten liegen würden.

Nun gibt es zwei Möglichkeiten:

(i) Die Fehler liegen auf verschiedenen Kontrollbit-Diagonalen. In diesem Fall sehen wir sofort wo die Fehler sind, nämlich dort wo die Kontroll-Diagonalen die fehlerhaften Spalten und Zeilen kreuzen.



(ii) Die Fehler liegen auf der selben Kontroll-Diagonale. Die Fehler heben sich auf dieser Diagonale auf und keine Kontroll-Diagonale zeigt einen Fehler an. Wir wissen somit auch eindeutig auf welchen zwei der vier gelben Schnittpunkte die Fehler liegen, da es nur eine Kontroll-Diagonale gibt die zwei Schnittpunkte beinhaltet, siehe folgendes Beispiel:



In beiden Fällen können wir Fehler zuverlässig erkennen und korrigieren.

Aufgabe

Angenommen die zwei Fehler sind auf der selben Zeile.

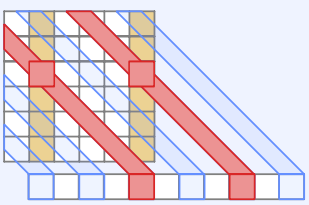
(1) Erklären Sie, warum mit Hilfe der zusätzlichen diagonalen Kontrollbits, beide Fehler genau lokalisiert werden können.

(2) Ist eine analoge Erklärung auch möglich, wenn die zwei Fehler stattdessen in der selben Spalte passieren?

Lösung

(1) Angenommen die zwei Fehler sind auf der selben Zeile. In diesem Fall ist die Summe der Einsen auf der fehlerhaften Zeile gerade, da sich die zwei Fehler gegenseitig aufheben. Aber die Spalten zeigen die zwei Fehler an.

Zudem wird jedes Bit in der fehlerbehafteten Zeile von genau einer anderen Diagonale gekreuzt. Aus diesem Grund haben bei zwei Fehlern genau zwei unterschiedliche Diagonalen eine ungerade Anzahl von Einsen und wir wissen somit auf welchen Diagonalen die Fehler auftreten.



Nun nehmen wir die Punkte, wo sich die fehlerhaften Diagonale (rot) mit den Fehlerhaften Spalten (gelb) schneiden um die Fehler zu lokalisieren.

(2) **Ja**, alle Argumente funktionieren analog zum vorderen Fall.

Da dies die einzigen möglichen Fälle sind, und in beiden Fällen zwei Fehler eindeutig erkannt werden können, funktioniert Franziskas Idee für alle möglichen Nachrichten.

Feststellung

Beim Kartentrick mit zusätzlichen diagonalen Kontrollbits lassen sich bis zu zwei Fehler erkennen und korrigieren.

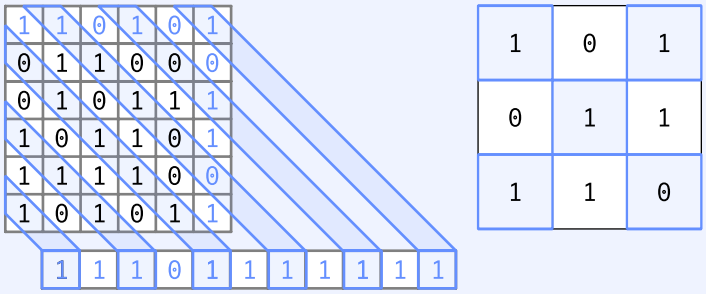
2.1.3. Wie sieht es mit drei Fehlern aus?

Zwei Fehler zu erkennen ist schon ein Schritt näher an einer Kodierung, die beliebig viele Fehler erkennen kann. Aber in der Praxis reicht das leider nicht immer aus. Wir möchten dewegen gerne untersuchen, ob unser Kartentrick beliebig erweitert werden kann. Dafür schauen wir uns jetzt denn Fall von drei Fehlern an.

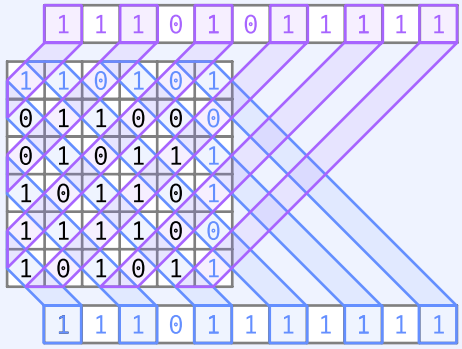
Aufgabe

Welche dieser Erweiterungen erkennen drei Fehler?

(a) Wir kombinieren den einfachen Kartentrick mit dem Ansatz von Ella und von Franziska. Sprich, wir berechnen zeilen- und spaltenweise Kontrollbits, diagonale Kontrollbits, und Kontrolbits für jedes 2x2 Feld. Das kann zum Beispiel wie folgt aussehen:



(b) Wir wenden den Trick von Franziska zweimal an. Sprich, wir berechnen zusätzlich zu den spalten-, zeilenweisen und diagonalen Kontrollbits, nochmals extra Kontrollbits für die anderen Diagonalen. Das sieht dann wie folgt aus:



Bilden Sie Gruppen von je zwei Personen. Eine Person baut drei Fehler ein, die andere Person versucht diese drei Fehler zu finden und zu korrigieren.

Finden sie dabei heraus, ob und welche der folgenden Schemas eine Selbstkorrektur von 3 Fehlern erlauben.

Lösung

(a) Wir schaffen es Gegenbeispiele zu finden. Ein solches Gegenbeispiel bekommen wir zum Beispiel, wenn wir drei Fehler direkt untereinander auf einer beliebigen Spalte einbauen.
(b) Wir schaffen es nicht Gegenbeispiele zu finden. Deswegen werden wir uns dieses Schema im folgenden genauer anschauen.

Mit zwei Diagonalen konnten wir bis zu drei Fehlern erkennen und korrigieren. Nun stellt sich die Frage, wie viele Extrabits wir für beliebig lange Nachrichten brauchen und ob das ganze System auch wirklich immer funktioniert.

Aufgabe

Angenommen die Karten sind quadratisch angeordnet und kodieren 2^m Nachrichten. Welche Länge hat die obige 3-fehlerkorrigierende Kodierung, bei welcher die Diagonalen in beide Richtung von Kontrollkarten überwacht werden?

Lösung

Das Quadrat hat Länge $\sqrt{m}$. Somit werden $6\sqrt{m}$ Kontrollkarten gelegt. Die Länge der 3-fehlerkorrigierenden Kodierung ist also $m + 6\sqrt{m}$.

Die Kodierung mit zusätzlichen Kontrollkarten für die weiteren Diagonalen können wir auf die 3-Fehlerkorrektur hin analysieren, indem wir wieder verschiedene Fälle anschauen.

Aufgabe

Wo können drei Fehler in Relation zueinander auftreten? Ergänzen Sie die folgende Einteilung in verschiedene Kategorien.

A: Alle drei Fehler sind auf einer Zeile

B: Zwei von drei Fehlern …
C: …

Lösung

Eine mögliche Einteilung ist die folgende:
- A: Alle drei Fehler sind auf einer Zeile.
- B: Zwei von drei Fehlern sind auf einer Zeile:
  - a: Der dritte Fehler ist in der selben Spalte wie einer der anderen beiden Fehler.
  - b: Der dritte Fehler ist in keiner der Spalten der ersten zwei Fehler.
- C: Alle drei Fehler sind auf verschiedenen Zeilen:
  - a: Alle Fehler sind auf verschiedenen Spalten.
  - b: Zwei Fehler sind auf der selben Spalte.
  - c: Alle Fehler sind auf der selben Spalte.

Wir sehen, für die Verteilung von drei umgedrehten Karten gibt es bereits bedeutend mehr Möglichkeiten. Somit gibt es auch bei wesentlich mehr Fällen von Fehlern zu prüfen, dass diese tatsächlich korrigiert werden.

Die obige Fallunterscheidung ist aber trotzdem nützlich um für gegebene Nachrichten Fehler zu finden und direkt zu korrigieren.

Falls wir aber nur zeigen wollen, dass die Kodierung 3-fehlerkorrigierend ist, erinnern wir uns an den Hemming Abstand. Wir wissen, dass eine 3-fehlerkorrigierende Kodierung mindestens einen Abstand von 7 benötig.

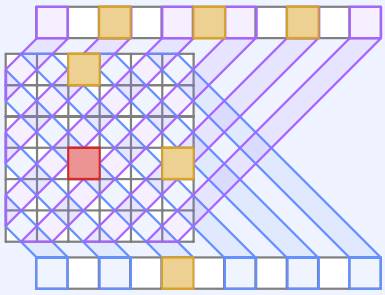
Aufgabe

Wir schauen uns Nachrichten mit einer unterschiedlichen Stelle an.
- (i) Wähle dafür eine beliebige Nachricht und trage alle Kontrollbits ein.
- (ii) Ändere einen Bit in der vorher gewählten Nachricht, und berechne alle Kontrollbits neu.
- (iii) Zähle wie viele Bits sich in Summe geändert haben.

Bonusaufgabe: Wiederhole das ganze mit Nachrichten, die sich in jeweils 2, 3, 4, 5, 6 und 7 Stellen unterscheiden. Zeige, dass auch hierbei mindestens 7 bits in Summe unterschiedlich sind.

Lösung

Wir wählen eine beliebige Nachricht und ändern einen Bit. Dieses ist rot markiert. Ing elb sind alle Kontrollbits markiert, die sich zusätzlich ändern.



Wir sehen, dass in Summe 7 Bits unterschiedlich sind:

- ein Bit in der Nachricht selbst (rot)
- zwei Bits vom einfachen Kartentrick (=Zeile/Spalte)
- zwei Bits bei den Diagonalen, die den geänderten Punkt in der Nachricht überwachen
- zwei weitere Bits in den Diagonalen, die die Kontrollbits des einfachen Kartentricks überwachen
- keine weiteren Bits auf den blauen Diagonalen, da dort beide Kontrollbits aus dem urspünglichen Kartentrick auf der selben Diagonale liegen und sich aufheben.
  Anmerkung: Würde der Fehler in der Nachricht an einer anderen Stelle liegen, könnten wir sogar noch zwei weitere unterschiedliche Bits bekommen. Schieben wir in diesem Beispiel den Fehler eine Zeile nach oben, so liegen die Kontrollbits aus dem urspünglichen Kartentrick nicht mehr auf der selben blauen Diagonale und heben sich somit auch nicht mehr auf.

Somit haben wir bei Nachrichten, die sich in einem Bit unterscheiden, einen Abstand von 7.

Bonusaufgabe: keine detaillierte Lösung hier.

Feststellung

Eine Erweiterung des Kartentricks für drei Fehler ist grundsätzlich möglich. Die Begründung ist aber aufwändig.

Bei einer Erweiterung auf vier oder mehr Fehler wird alles nochmals umfangreicher und umständlicher. Aber zum Glück gibt es nebst dem Kartentrick noch weitere Kodierungen. Im nächsten Abschnitt werden wir ohne viel Aufwand systematisch eine k-fehlerkorrigierende Kodierung für beliebiges $k \geq 0$ entwickeln.

2.2. Reed-Solomon

Die bisher betrachteten Kartentricks sind Fehlerkodierungen, welche einen, zwei bzw. drei Fehler erkennen und korrigieren können. Bereits für den zuletzt gesehenen Kartentrick ist die Begründung, dass er immer 3-fehlerkorrigierend ist, allerdings sehr aufwändig.

In diesem Abschnitt wollen wir uns eine Version des sogenannten Reed-Solomon Verfahrens anschauen, welches eine k-fehlerkorrigierende Kodierung für beliebiges $k \geq 0$ bietet und mit einer wenig aufwändigen Begründung.

Diese Kodierung wurde ab 1960 entwickelt und das erste mal 1977 auf dem Voyager 1 Satelliten angewandt, der noch heute am Rande des Sonnensystems für uns Daten sammelt.

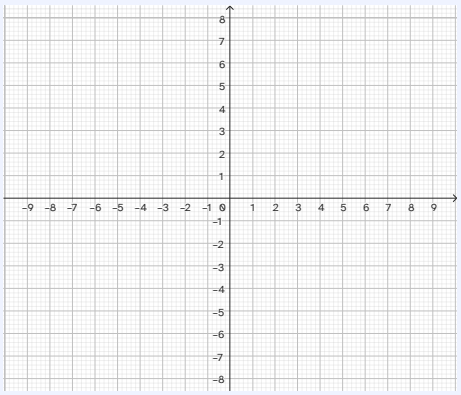
Das Reed-Solomon Verfahren verwendet Polynomgleichungen. Wir wollen eine Grundidee davon kennenlernen und beschränken uns dabei auf lineare Gleichungen.

2.2.1. Auffrischung zu linearen Gleichungen

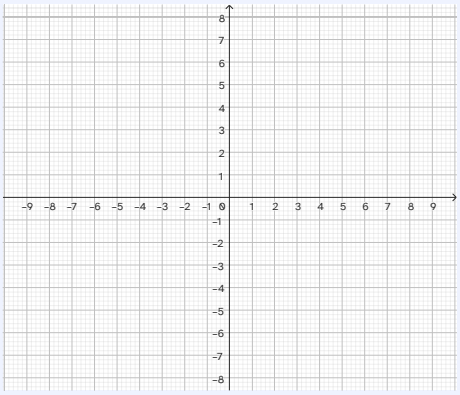
Wir betrachten lineare Gleichungen der Form $y = mx + b$. Ihre Lösungsmenge ist eine nicht vertikale Gerade in der Koordinatenebene, welche Steigung m hat und die y-Achse im Punkt $(0, b)$ schneidet.

Aufgabe

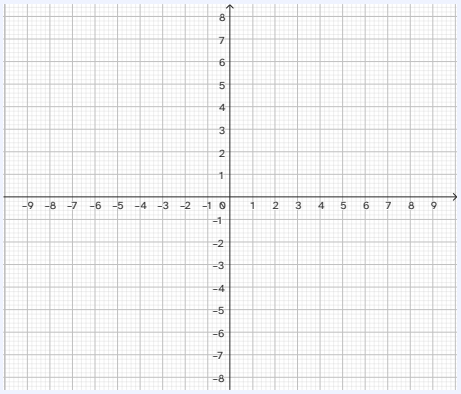
Zeichnen Sie für folgende linearen Gleichungen die zugehörige Gerade in die Koordinatenebene ein. Bestimmen Sie jeweils die Steigung m und den Achsenabschnitt b.

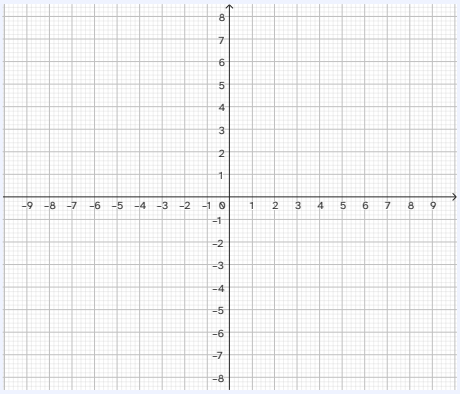


$y = 3x - 2$



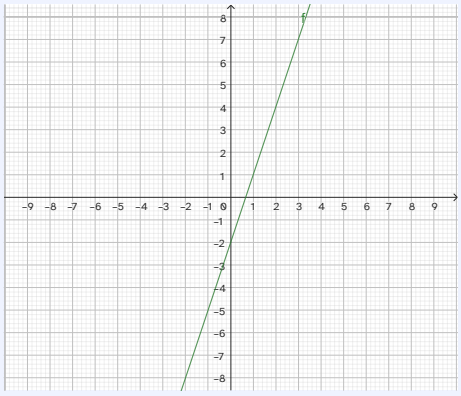
$y = \frac{3}{2}x + 1$



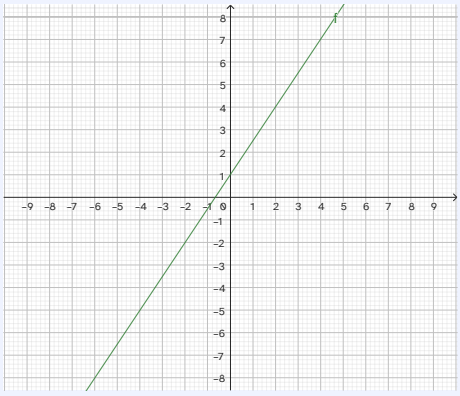


$y = 3$ $y = -2x$

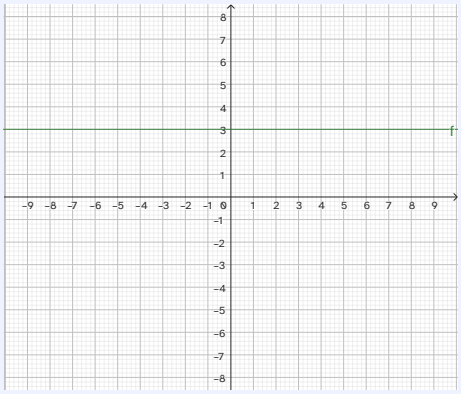
Lösung



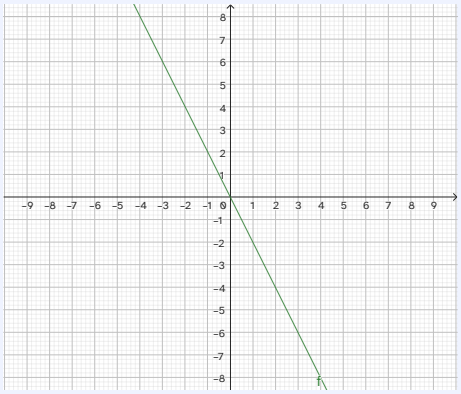
Für $y = 3x - 2$ ist $m = 3$ und $b = -2$.



Für $y = \frac{3}{2}x + 1$ ist $m = \frac{3}{2}$ und $b = 1$



Für $y = 3$ ist $m = 0$ und $b = 3$

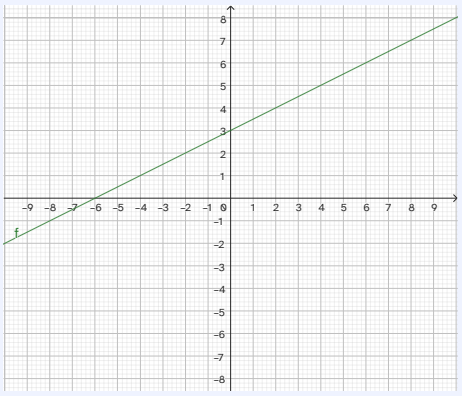


Für $y = -2x$ ist $m = -2$ und $b = 0$

Umgekehrt ist jede nicht vertikale Gerade in der Ebene die Lösungsmenge einer eindeutigen linearen Gleichung.

Aufgabe

Bestimmen Sie die lineare Gleichung mit folgender Gerade als Lösungsmenge:



Lösung

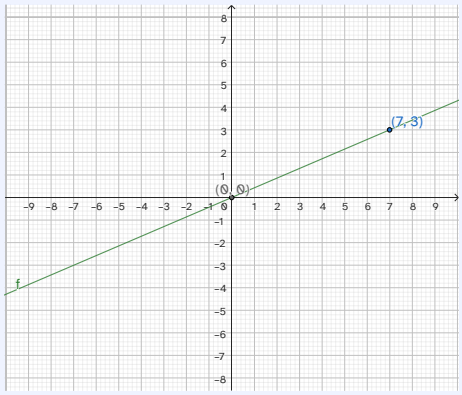
Die Gerade schneidet die y-Achse in $(0, 3)$. Somit ist der Achsenabschnitt $b = 3$. Die Verschiebung um 2 in x-Richtung bedeutet für die Punkte auf der Geraden eine Verschiebung um 1 in y-Richtung. Somit ist die Steigung m der Geraden gleich $\frac{1}{2}$. Somit ist $y = \frac{1}{2}x + 3$ die gesuchte lineare Gleichung.

Aufgabe

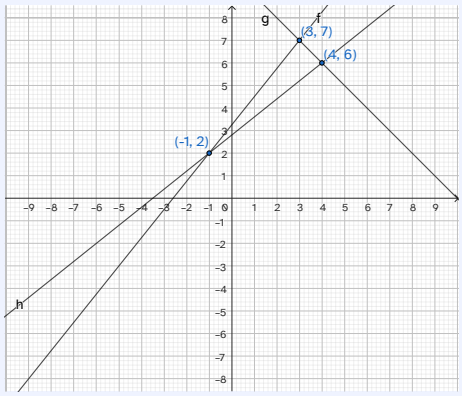
1. Wieviele Geraden gibt es, welche durch die Punkte $P_1 = (0, 0)$ und $P_2 = (7, 3)$ gehen? Zeichnen Sie diese Geraden ein und bestimmen Sie jeweils die Steigung und den Achsenabschnitt.
2. Bestimmen Sie einen dritten Punkt, welcher auf der Geraden der vorherigen Aufgabe liegt.
3. Wieviele Geraden gibt es, welche durch die Punkte $P_1 = (-1, 2)$, $P_2 = (4, 6)$ und $P_3 = (3, 7)$ gehen? Wieviele Geraden gibt es, welche durch zwei dieser drei Punkte gehen?

Lösung

1. Durch zwei Punkte in der Ebene geht jeweils genau eine Gerade, also auch im Fall der Punkte $(0, 0)$ und $(7, 3)$. Die Gerade durch $(0, 0)$ und $(7, 3)$ schneidet die y-Achse im Punkt $(0, 0)$. Somit ist $(0, b) = (0, 0)$, also $b = 0$. Die Steigung der Gerade ist $m = \frac{3-0}{7-0} = \frac{3}{7}$.



2. Die lineare Gleichung zu obiger Gerade ist also $y = \frac{3}{7}x$. Zum Beispiel erfüllt der Punkt $\left(1, \frac{3}{7}\right)$ diese Gleichung und liegt folglich auch auf der Geraden.
3. Durch je zwei der Punkte geht jeweils eine eindeutige Gerade. Diese zeichnen wir jeweils ein und sehen, dass insgesamt drei Geraden durch zwei dieser Punkte gehen, aber keine Gerade durch alle drei Punkte geht.



Feststellung

Wir können eine lineare Gleichungen auf drei Arten eindeutig beschreiben und somit bestimmen:

1. Durch Gleichungskoeffizienten m und b der linearen Gleichung $y = mx + b$.
2. Zeichnerisch durch eine nicht vertikale Gerade in der Koordinatenebene.
3. Durch zwei Punkte in der Ebene, welche nicht vertikal zueinander stehen.

2.2.2. Übermittlung von Punkten auf einer Geraden

Die Architektin Aline zeichnet in die Koordinatenebene eine nicht vertikale Gerade ein und möchte Ingenieurin Ines über diese Gerade elektronisch informieren.

Aufgabe

Aline möchte dafür keine Bilddatei versenden. Welche Angaben könnte Aline stattdessen an Ines übermitteln, so dass Ines die Gerade dennoch rekonstruieren kann?

Lösung

Aline könnte Ines die Steigung m zusammen mit dem Achsenabschnitt b angeben. Stattdessen könnte sie aber auch die Koordinaten von zwei beliebigen unterschiedlichen Punkten auf der Geraden angeben. In beiden Fällen bestimmen diese Angaben die Gerade eindeutig und erlauben die Rekonstruktion der Geraden.

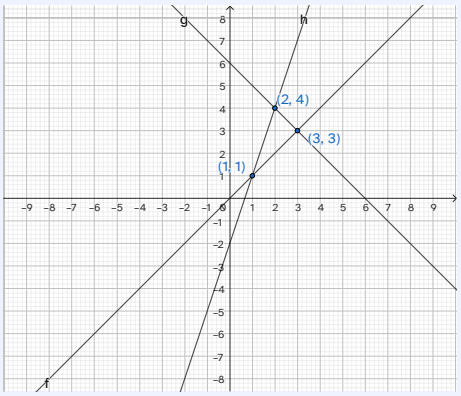
Aline entscheidet sich, Ines die Koordinaten von zwei Punkten auf der Geraden zu übermitteln. Weil manchmal bei der elektronischen Uebermittlung Fehler passieren, schickt Aline sicherheitshalber die Koordinaten von mehr als zwei Punkten, welche auf der Geraden liegen.

Aufgabe

Zunächst erhält Ines von Aline die drei Punkte $(1, 1)$ und $(2, 4)$ und $(3, 3)$ übermittelt. Kann Ines entscheiden, ob die Übermittlung fehlerlos war?

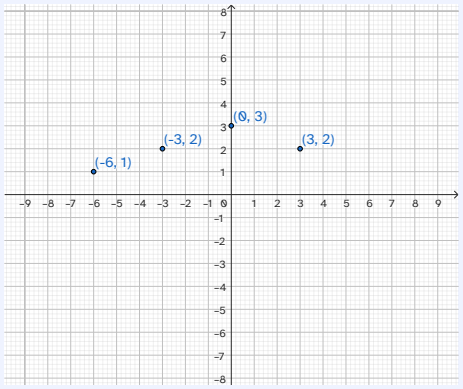
Lösung

Die drei erhaltenen Punkte liegen nicht auf einer Geraden. Weil die von Aline gewählten Punkte aber auf einer Gerade liegen, kann Ines folgern, dass bei der Übermittlung mindestens ein Fehler passiert ist.



Aufgabe

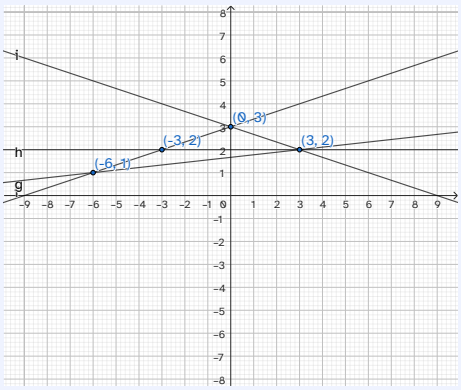
Nun erhält Ines vier Punkte von Aline. Diese zeichnet Ines wie folgt in ein Koordinatensystem:



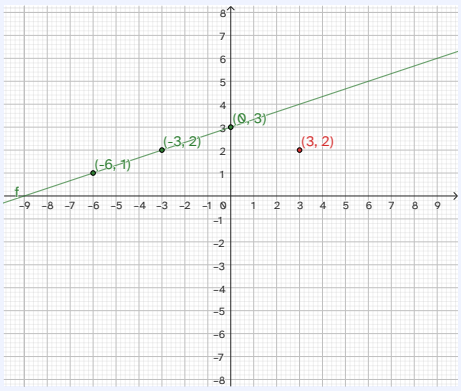
Ines nimmt an, dass mindestens drei der vier Punkte fehlerlos übermittelt wurden. Kann Ines unter dieser Annahme die Gerade von Aline nachzeichnen?

Lösung

Wir zeichnen zuerst alle Geraden ein, welche durch mindestens zwei der vier Punkte gehen:

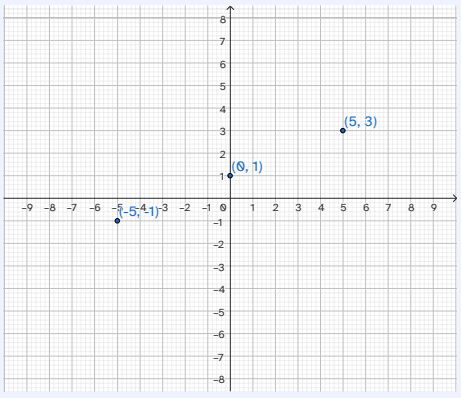


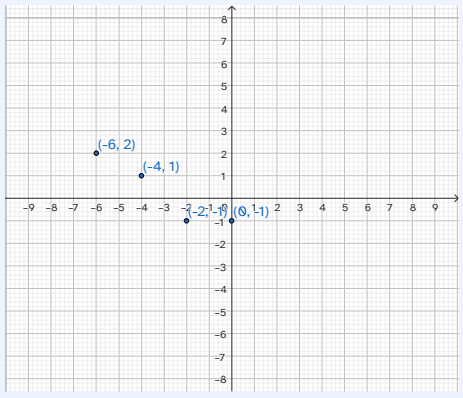
Weil nach Annahme drei Punkte auf der Geraden von Aline richtig übermittelt wurden, müssen diese immer noch auf dieser Geraden liegen. Folgende Gerade ist die einzige Gerade, welche durch drei der Punkte geht, und muss deshalb die Gerade von Aline sein:

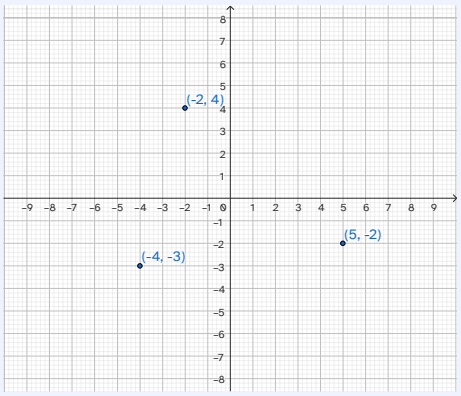


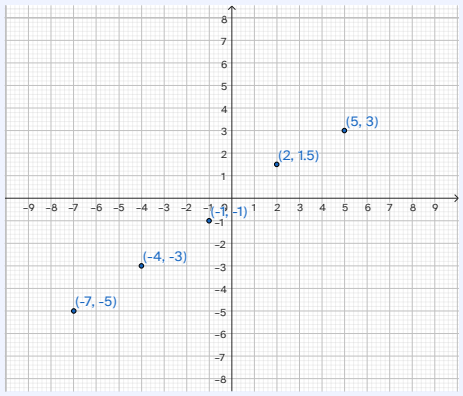
Aufgabe

Aline schickt Ines für vier weitere Geraden jeweils Punkte. Ines zeichnet diese jeweils wie folgt ein:



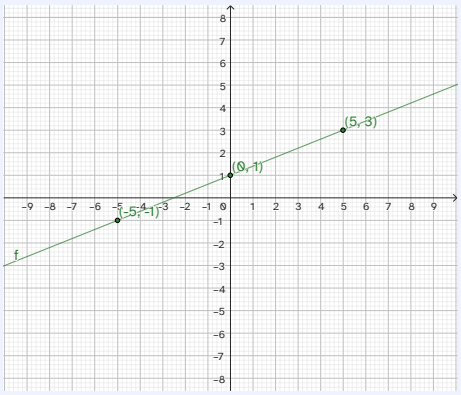


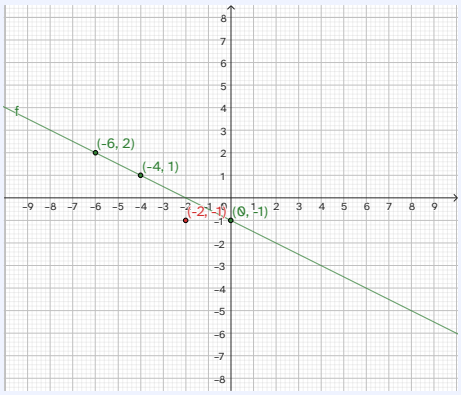


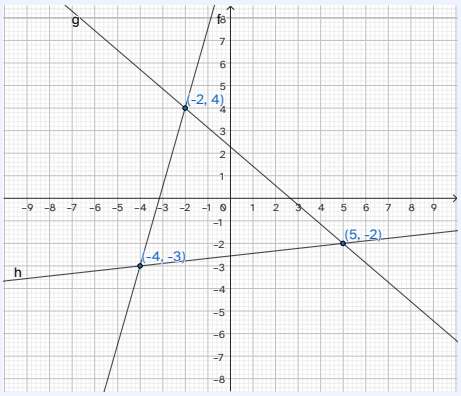


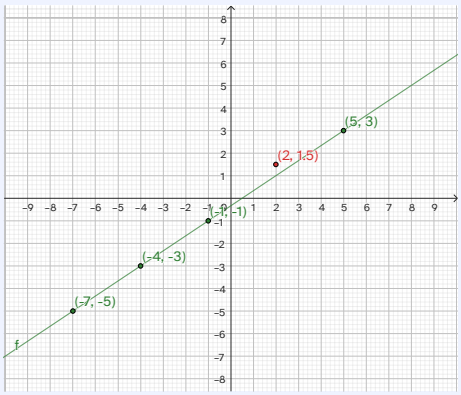
Angenommen, höchstens ein Punkt wurde jeweils fehlerhaft übermittelt. Kann Ines dann jeweils bestimmen, ob ein Fehler passiert ist? Bestimmen Sie jeweils gegebenenfalls den fehlerhaft übermittelten Punkt.

Lösung









Unter der Annahme, dass höchstens ein Punkt fehlerhaft übermittelt wird, genügt also die Übermittlung von 4 Punkten der Geraden, um die Gerade zu rekonstruieren.

Aufgabe

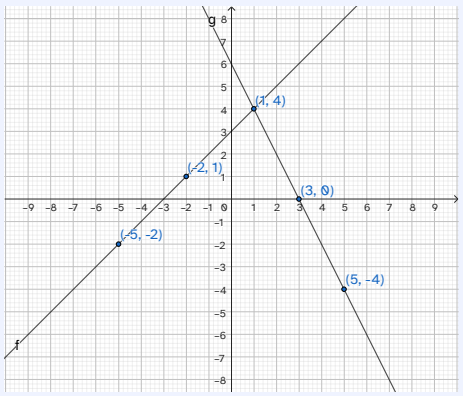
Aline und Ines fragen sich, ob die Übermittlung von 5 Punkten der Gerade zur Rekonstruktion genügt, falls zwei dieser Punkte fehlerhaft übermittelt werden. Falls die Antwort auf diese Frage ja ist, müsste die Begründung aus einem

Beweis, also einer allgemein gültigen Erklärung bestehen. Falls die Antwort nein ist, genügt als Begründung ein *Gegenbeispiel*, also ein Beispiel bestehend aus 5 übermittelten Punkten, wovon zwei fehlerhaft übermittelt worden sind, aus denen sich die Gerade nicht rekonstruieren lässt.

Können Sie die Frage beantworten mit einer Begründung?

Lösung

Die korrekte Übermittlung von 3 von 5 Punkten genügt im Allgemeinen nicht, um die Gerade zu rekonstruieren. Wie folgendes Beispiel zeigt, können die übermittelten 5 Punkte so liegen, dass zwei verschiedene Geraden durch je drei der 5 Punkte gehen, womit sich nicht entscheiden lässt, welches die Gerade von Aline ist.

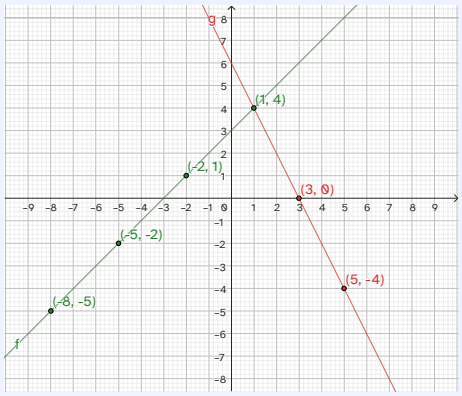


Aufgabe

Genügt die Übermittlung von 6 Punkten der Gerade zur Rekonstruktion der Gerade, falls zwei dieser Punkte fehlerhaft übermittelt werden?

Lösung

Wenn 6 Punkte übermittelt werden und höchstens 2 Punkte falsch übermittelt worden sind, müssen mindestens 4 der übermittelten Punkte auf der gesuchten Geraden liegen. Andererseits existiert höchstens eine Gerade, welche durch 4 von 6 Punkten geht: Tatsächlich kann eine weitere Gerade höchstens einen dieser vier Punkte enthalten, womit sie ingesamt höchstens 3 der 6 Punkte enthalten könnte. Die gesuchte Gerade ist also die eindeutige Gerade, auf welcher mindestens 4 der 6 Punkte liegen, und kann somit rekonstruiert werden.



Wir nehmen ab jetzt an, dass jeweils höchstens k Punkte fehlerhaft übermittelt werden, wobei k eine beliebige natürliche Zahl ist. Aline und Ines fragen sich, wie viele n Punkte Aline unter dieser Annahme übermitteln muss, damit Ines die Gerade dennoch rekonstruieren kann. Die Annahme ist gleichbedeutend dazu, dass mindestens $n - k$ der übermittelten Punkte korrekt übermittelt werden, also nach der Übermittlung auf der gesuchten Gerade liegen.

Wenn nur eine Gerade existierte, welche mindestens $n - k$ von den n Punkten enthält, dann müsste dies also die gesuchte Gerade sein, und diese könnte somit rekonstruiert werden. Die Frage ist also: Wie gross muss n mindestens sein, damit höchstens eine Gerade durch $n - k$ von den n Punkten gehen kann?

Wir betrachten zuerst den gegenteiligen Fall, wo also zwei verschiedene Geraden existieren, welche je mindestens $n - k$ von den n Punkten enthalten.

Auf beiden Geraden liegen also je $n - k$ Punkte der n Punkte. Da sich die Geraden nur in einem Punkt schneiden, enthalten sie höchstens einen der n Punkte gemeinsam. Beide Geraden zusammen enthalten also mindestens $(n - k) + (n - k) - 1 = (2n - 2k - 1)$ der n Punkte. In diesem Fall muss also $2n - 2k - 1 \leq n$ sein, das heisst $n \leq 2k + 1$. Umgekehrt bedeutet dies, dass für $n \geq 2k + 2$ keine zwei verschiedene Geraden existieren, welche je $n - k$ Punkte von den n Punkten enthalten.

Feststellung

Fall höchstens k Punkte fehlerhaft übermittelt werden, lässt sich aus $n \geq 2k + 2$ übermittelten Punkten die Gerade rekonstruieren,

Aufgabe

Angenommen, wir können höchstens 24 auf einer Geraden liegende Punkte übermitteln. Wieviele Fehler dürfen dabei höchstens passieren, damit diese Gerade aus den übermittelten Punkten dennoch rekonstruiert werden kann?

Lösung

Hier ist $n = 24$ und gesucht ist die Anzahl k der Fehler, die wir höchstens machen dürfen. Wie wir allgemein hergeleitet haben, muss $2k + 2 \leq 24$ sein, also $k \leq 11$.

Sie hatten durch ein Gegenbeispiel gesehen, dass 5 übermittelte Punkte für die Rekonstruktion nicht genügen, wenn 2 Punkte davon fehlerhaft übermittelt wurden. Dabei war $n = 5$, $k = 2$ und also $n = 2k + 1$. Mit der gleichen Idee können wir für allgemeine n und k, so dass $n \leq 2k + 1$ ist, n Punkte in der Ebene finden, so dass zwei verschiedene Geraden je durch $n - k$ dieser n Punkte gehen.

Aufgabe

Finden Sie ein solches Gegenbeispiel für $n = 9$ und $k = 4$.

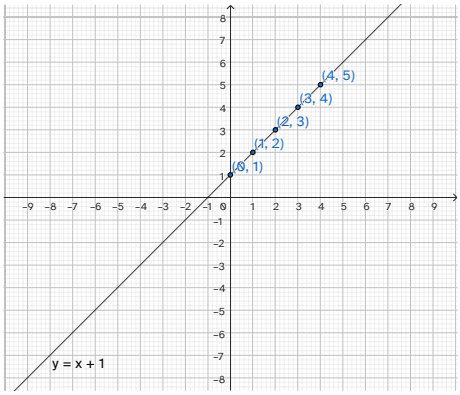
Feststellung

Falls k Punkte fehlerhaft übermittelt werden, lässt sich aus $n = 2k + 1$ übermittelten Punkten die Gerade nicht immer rekonstruieren.

2.2.3. Überblick zum Reed-Solomon Verfahren

Das Reed-Solomon-Verfahren erfolgt in folgenden Schritten, welche Sie danach sogleich in Beispielen genauer kennenlernen. Dabei ist $n \geq 2$ eine natürliche Zahl.

1. Eine zu übermittelnde Information, welche typischerweise als Bitfolge vorliegt, wird zuerst durch zwei Zahlen m und b im Dezimalzahlensystem dargestellt.
2. Danach werden die Werte $y_i = mi + b$ für alle $i = 0, \ldots, n - 1$ berechnet. Dies sind genau die y-Koordinaten der n Punkte (i, y_i) auf der Geraden von $y = mx + b$. Im folgenden Bild ist $n = 5$ und $y_i = i + 1$ für alle $i = 0, \ldots, 4$.



3. Übermittelt werden $y_0, \ldots, y_{n-1}$ in dieser Reihenfolge und in Binärdarstellung. Der Empfänger weiss, dass dies die y-Koordinaten in Binärdarstellung sind von Punkten mit x-Koordinaten $0, \ldots, n - 1$.
4. Werden diese n y-Koordinaten fehlerlos übermittelt, lässt sich aus den n Punkten $(0, y_0), (1, y_1), \ldots, (n - 1, y_{n-1})$ in Dezimaldarstellung die Gerade (und also m und b) rekonstruieren, denn durch zwei oder mehr vorgegebene Punkte geht höchstens eine Gerade.
5. Die erhaltenen m und b werden nun wieder in ihre anfängliche Form übersetzt, also typischerweise in eine Bitfolge.

Feststellung

Unter der Annahme, dass im dritten Schritt höchstens k der y-Koordinaten falsch übermittelt werden, reicht die Übermittlung von $n = 2k + 2$ y-Koordinaten von

Punkten einer Gerade, um die Gerade aus den übermittelten Koordinaten zu rekonstruieren. Mit anderen Worten ist die Kodierung von Geraden durch $2k+2$ y-Koordinaten von enthaltenen Punkten k-fehlerkorrigierend.

Wir wollen die Schritte im Reed-Solomon Verfahren in einem Beispiel nachvollziehen.

Aufgabe

Wir wollen die Bitfolge 00110001 verschicken. Wie könnte diese Bitfolge als Steigung m und Achsenabschnitt b dargestellt werden?

Lösung

Eine Möglichkeit ist, die Bitfolge zuerst in der Mitte zu trennen: 0011 | 0001 und dann vom Binärsystem zum Dezimalsystem zu wechseln: 0011 -> 3 = m und 0001 -> 1 = b. Dann ist die Steigung $m = 3$ und der Achsenabschnitt $b = 1$.

Aufgabe

Wieviele n Punkte müssen mindestens übermittelt werden für eine 2-fehlerkorrigierende Kodierung? Bestimmen Sie diese Punkte. Die x-Koordinaten dabei sollen in $\{0, 1, 2, 3, .., n-1\}$ sein.

Lösung

Die Mindestanzahl zu übermitteltender Punkte ist $2k + 2 = 6$. Für $m = 3$ und $b = 1$ bestimmen wir die Punkte

1. $x = 0; y = 3x + 1 = 1; P_1 = (0, 1)$
2. $x = 1; y = 3x + 1 = 4; P_1 = (1, 4)$
3. $x = 2; y = 3x + 1 = 7; P_1 = (2, 7)$
4. $x = 3; y = 3x + 1 = 10; P_1 = (3, 10)$
5. $x = 4; y = 3x + 1 = 13; P_1 = (4, 13)$
6. $x = 5; y = 3x + 1 = 16; P_1 = (5, 16)$

Nun sollen die y-Koordinaten 1, 4, 7, 10, 13, 16 in binärer Darstellung verschickt werden.

Aufgabe

Stellen Sie die Zahlen im Binärsystem dar und bilden Sie daraus ein Codewort, so dass der Empfänger danach die Dezimaldarstellung der Zahlen damit berechnen kann.

Lösung

Weil die höchste Zahl im Beispiel 16 ist, werden 5 Bits pro Zahl gebraucht. Somit ergeben sich folgende Binärdarstellungen der Zahlen:

1. 1 = 00001

2. 4 = 00100
3. 7 = 00111
4. 10 = 01010
5. 13 = 01110
6. 16 = 10000

Somit ist 00001|00100|00111|01010|01110|10000 ein Codewort, welches die Umwandlung zurück in die Dezimaldarstellung ermöglicht .

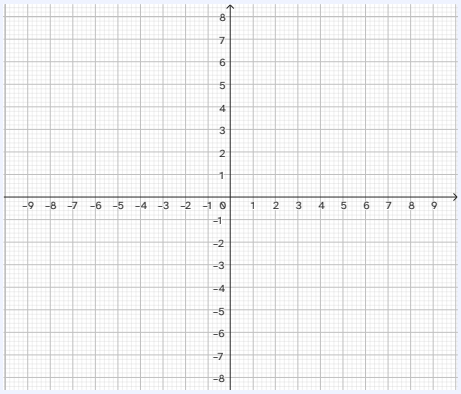
Nun schickt Aline das Codewort zu Ines. Ines erhält das folgende Codewort:

$$000010010100111010100101010000$$

Ines weiss, dass jede Koordinate in 5 Bits dargestellt ist.

Aufgabe

Zeichnen Sie das Codewort im Koordinatensystem ein und bestimmen Sie ob Fehler passiert sind und, wenn ja, wo. Sie können annehmen, dass maximal zwei Fehler geschehen sind.



Lösung

Im Wissen, dass jede Koordinate in 5 Bits dargestellt ist, teilen wir die Bitfolge auf: `00001|00101|00111|01010|11110|10000`

Nun rechnen wir ins Dezimalzahlensystem um

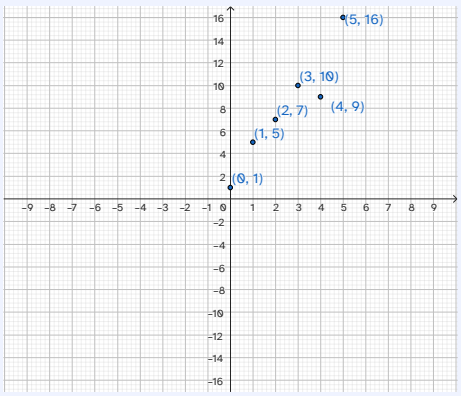
1. `00001` = 1
2. `00101` = 5
3. `00111` = 7
4. `01010` = 10
5. `01010` = 9
6. `10000` = 16

Bekannt ist auch, welche x-Koordinate zu welcher y-Koordinate gehört. Somit erhalten wir die Punkte

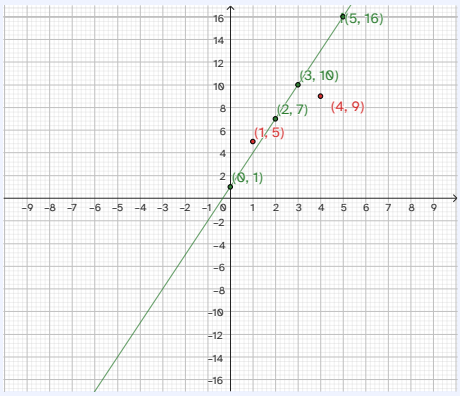
1. $P_1 = (0, 1)$
2. $P_2 = (1, 5)$
3. $P_3 = (2, 7)$

4. $P_4 = (3, 10)$
5. $P_5 = (4, 9)$
6. $P_6 = (5, 16)$

und zeichnen diese ins Koordinatensystem ein:



Unter der Annahme, dass maximal 2 Fehler passiert sind, muss genau eine Gerade durch mindesten vier dieser Punkte geht, nämlich folgende Gerade:



Aufgabe

Bestimmen Sie mit Hilfe der vorherigen Lösung die versandte Nachricht.

Lösung

Wir müssen von den korrekten Punkten die Steigung und den Achsenabschnitt bestimmen. Durch die grafische Darstellung wissen wir, dass Punkt P_2 und Punkt P_5 fehlerhaft übermittelt wurden. Somit können wir mit einem beliebigen Paar der anderen Punkte die Steigung $m = 3$ und den Achsenabschnitt $b = 1$ bestimmen. Nun übersetzen wir wieder $3 \rightarrow 0011$ und $1 \rightarrow 0001$ und erhalten 00110001.

Das betrachtete Reed-Solomon Verfahren unterscheidet sich vom Kartentrick insbesondere in folgenden Punkten:

- Bei Reed-Solomon ist die Nachricht, also m und b in Binärdarstellung, nicht Teil des übermittelten Code-Worts. Das Code-Wort setzt sich aus den Binärdarstellung der y-Koordinaten gewisser Punkte zusammen.

- Der Kartentrick korrigiert jeweils das Code-Wort. Bei Reed-Solomon ist das auch möglich, aber nicht nötig, um m und b zu rekonstruieren. Tatsächlich reichen bereits genügend viele korrekt übermittelte y-Koordinaten, um die Gerade und folglich ihre Steigung m und ihren Achsenabschnitt b zu ermitteln.

Dennoch könnten wir eine falsch übermittelte y-Koordinate y_i unter Kenntnis der Geraden korrigieren, weil wir ihre x-Komponenten i kennen: nämlich muss dann $y_i = mi + b$ sein. Die Korrektur der y-Komponenten entspricht der Korrektur des Code-Worts.

Bei der vorherigen Aufgabe ist das schnell ersichtlich:

Verschickte Nachricht: 00001|00100|00111|01010|01110|10000

Erhaltene Nachricht: 00001|00101|00111|01010|11110|10000

Wir ermittelten $m = 3$ und $b = 1$, ohne die Bits der erhaltenen Nachricht zu korrigieren, sondern indem wir die eindeutige Gerade bestimmt haben, welche durch mindestens vier der sechs Punkte geht. Das Code-Wort können wir dennoch korrigieren, indem wir $y_1 = 3 \cdot 1 + 1 = 4$ und $y_4 = 3 \cdot 4 + 1 = 13$ berechnen, diese beiden Zahlen in je 5 Bits binär darstellen und mit diesen Bitfolgen die erhaltene Nachricht bei der zweiten und fünften Bitfolge korrigieren.

Aufgabe

Der Korrektur von wievielen Bits maximal entspricht die Korrektur von k y-Koordinaten, wenn wir pro y-Koordinate 5 Bits benötigen?

Lösung

Die Korrektur von einer y-Koordinate entspricht der Korrektur von maximal 5 Bits. Folglich entspricht die Korrektur von k y-Koordinaten der Korrektur von maximal $5 \cdot k$ Bits.

Feststellung

Das betrachtete Reed-Solomon Verfahren wandelt die Nachricht in ein Code-Wort um, welches die Nachricht nicht enthält.

Die Fehlerkorrektur zielt nicht darauf ab, einzelne Bits zu korrigieren, sondern genügend korrekt übermittelte Punkte zu finden, um mit diesen die Gerade und somit deren Steigung und Achsenabschnitt zu bestimmen.

Zuletzt vergleichen wir die betrachtete Reed-Solomon Kodierung mit der Kodierung, welche n Kopien der ursprünglichen Bitfolge übermittelt.

Falls zum Beispiel $n = 4$ ist und wiederum 00110001 die ursprüngliche Bitfolge ist, dann wird mit letzterer Kodierung also

$$00110001|00110001|00110001|00110001$$

verschickt.

Aufgabe

Wieviele Bits dürfen bei der Übermittlung von 4 Kopien einer zu übermittelnden Bitfolge höchstens fehlerhaft übermittelt werden, damit die Bitfolge dennoch rekonstruiert werden kann?

Lösung

Falls höchstens ein Bit fehlerhaft übermittelt wird, werden mindestens drei der vier Kopien der Bitfolge korrekt übermittelt. Insbesondere sind dann mindestens drei der vier erhaltenen Kopien gleich, nämlich gleich der ursprünglichen Bitfolge. Andererseits existiert höchstens eine Bitfolge, welche mit mindestens drei der vier Kopien übereinstimmt. Die ursprüngliche Bitfolge ist also jene eindeutige Bitfolge, welche mit mindestens drei der vier Kopien übereinstimmt, und lässt sich somit rekonstruieren.

Falls zwei Bits fehlerhaft übermittelt werden, ist die Rekonstruktion nicht immer möglich. Wenn nämlich bei zwei der vier Kopien an der jeweils gleichen Stelle das Bit fehlerhaft übermittelt wird, dann enstehen jeweils zwei Kopien von zwei unterschiedlichen Bitfolgen, und welche der zwei unterschiedlichen Bitfolgen die ursprüngliche Bitfolge ist, lässt sich nicht entscheiden. Falls beispielsweise 00110001 die ursprüngliche Bitfolge ist und bei den letzten zwei Kopien jeweils das Bit an sechster Stelle fehlerhaft übermittelt wird, dann entsteht

$$00110001|00110001|00110101|00110101$$

Hier sind tatsächlich die ersten zwei Bitfolgen, bzw. die letzten zwei Bitfolgen, jeweils Kopien der Bitfolge 00110001, bzw. 00110101, womit nicht entschieden werden kann, welche der beiden Bitfolgen die ursprüngliche ist.

Die Kodierung durch 4 Kopien ist also 1-fehlerkorrigierend, aber nicht 2-fehlerkorrigierend. Bei dieser Kodierung darf also höchstens ein Bit fehlerhaft übermittelt werden, damit die Bitfolge dennoch rekonstruiert werden kann.

Aufgabe

Wie viele Kopien einer Bitfolge müssen übermittelt werden, damit die Kodierung 2-fehlerkorrigierend ist?

Lösung

Wie wir in der vorderen Aufgabe gesehen haben, ist die Kodierun durch 4 Kopien nicht 2-fehlerkorrigierend. Die Übermittlung von 5 Kopien ist aber 2-fehlerkorrigierend. Tatsächlich können zwei fehlerhaft übermittelte Bits in maximal zwei Kopien liegen, womit mindestens drei Kopien korrekt übermittelt werden, also gleich der ursprünglichen Bitfolge sein müssen. Andererseits existiert höchstens eine Bitfolge, welche mit mindestens drei der 5 Kopien übereinstimmt. Die ursprüngliche Bitfolge ist also jene eindeutige Folge, welche mit mindestens drei

der 5 Kopien übereinstimmt, und lässt sich somit rekonstruieren. Die Kodierung durch 5 Kopien ist also 2-fehlerkorrigierend.

Unter gewissen Voraussetzungen lässt sich die Reed-Solomon Kodierung mit der Kodierung durch Kopieren hinsichtlich Effizienz vergleichen:

Aufgabe

Eine Bitfolge der Länge 8 soll einmal mit der Reed-Solomon Kodierung und einmal mit der Kodierung durch Kopien übermitteln. Dabei nehmen wir an, dass die y-Koordinaten der Punkte beim Reed-Solomon Verfahren je mit 5 Bits dargestellt werden können, wie dies beim Beispiel 00110001 der Fall war. Wieviele Bits müssen bei diesen beiden Methoden jeweils mindestens übermittelt werden, damit die Übermittlung jeweils 2-fehlerkorrigierend ist? Welche der beiden Methode ist demnach effizienter?

Lösung

Wie wir gesehen haben, ist die Reed-Solomon Kodierung 2-fehlerkorrigierend, falls die y-Koordinaten von mindestens 6 Punkten übermittelt werden. Weil die Koordinaten nach Annahme je mit 5 Bits dargestellt werden können, ist die Reed-Solomon Kodierung somit bereits durch die Übermittlung von $6 \cdot 5$ Bits 2-fehlerkorrigierend.

Andererseits haben wir gesehen, dass die Kodierung mit n Kopien erst 2-fehlerkorrigierend ist für $n \geq 5$. Weil die ursprüngliche Bitfolge aus 8 Bits besteht, müssen bei dieser zweiten Kodierung also mindestens $5 \cdot 8$ Bits übermittelt werden, damit 2 Fehler korrigiert werden können.

Die Reed-Solomon Kodierung benötigt für eine 2-Fehlerkorrektur folglich weniger Bits als die Kodierung durch Kopieren und ist somit effizienter.

Feststellung

Unter der Voraussetzung, dass bei der Reed-Solomon Kodierung die zu übermittelnden y-Koordinaten mit wenigen Bits dargestellt werden können, ist die Reed-Solomon Kodierung effizienter als die Kodierung durch Kopieren.

Im betrachten Reed-Solomon Verfahren haben wir lineare Gleichungen, deren Koeffizienten reelle Zahlen sind, verwendet. Die Koordinaten der Punkte in der Lösungsmenge einer solchen linearen Gleichung bestehen ebenfalls aus reellen Zahlen. Bei den betrachteten Beispielen konnten wir die y-Koordinaten ausgewählter Punkte dabei jeweils mit wenigen Bits darstellen. Allgemein lassen sich reelle Zahlen aber nicht durch Bitfolgen kontrollierbarer, endlicher Länge darstellen. Insbesondere aus diesem Grund wird beim Reed-Solomon Verfahren oftmals der Zahlenbereich der reellen Zahlen durch einen sogenannten *endlichen Körper* ersetzt. Ein endlicher Körper besteht aus einer endlichen Menge zusammen mit einer Additions-, Subtraktions-, Multiplikations- und einer Divisionsoperation zwischen Elementen dieser Menge, so dass vom reellen

Zahlenbereich bekannte Rechengesetze für diese Operationen gelten. Weil wir mit den Elementen eines endlichen Körpers analog rechnen können wie mit reellen Zahlen, lässt sich das Reed-Solomon Verfahren analog mit einem endlichen Körper anstatt dem reellen Zahlenbereich durchführen. Weil ein endlicher Körper nur endlich viele Elemente enthält, genügen ausserdem nun endliche viele Bits, um all seine Elemente durch diese Bits darzustellen, wodurch das Reed-Solomon Verfahren auch effizient wird.