Algebra3, elemző szakirány

ELTE Algebra és Számelmélet Tanszék

Előadó: Kiss Emil
ewkiss@cs.elte.hu

10. előadás

A kódok típusai

Kódolás: adatok megváltoztatása.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- Titkosírás (kriptográfia).

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- **Titkosírás** (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- **Titkosírás** (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.
  Például az **RSA-módszer** azt az elvet alkalmazza, hogy nagy számok nem bonthatók gyorsan prímek szorzatára.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- **Titkosírás** (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.
  Például az **RSA-módszer** azt az elvet alkalmazza, hogy nagy számok nem bonthatók gyorsan prímek szorzatára.
- **Forráskódolás:** adatok tömörítése.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- **Titkosírás** (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.
  Például az **RSA-módszer** azt az elvet alkalmazza, hogy nagy számok nem bonthatók gyorsan prímek szorzatára.
- **Forráskódolás:** adatok tömörítése.
  Kevesebb tárolóhely,

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- **Titkosírás** (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.
  Például az **RSA-módszer** azt az elvet alkalmazza, hogy nagy számok nem bonthatók gyorsan prímek szorzatára.
- **Forráskódolás:** adatok tömörítése.
  Kevesebb tárolóhely, gyorsabb adattovábbítás.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- Titkosírás (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.
  Például az RSA-módszer azt az elvet alkalmazza, hogy nagy számok nem bonthatók gyorsan prímek szorzatára.
- Forráskódolás: adatok tömörítése.
  Kevesebb tárolóhely, gyorsabb adattovábbítás.
- Hibajelző és hibajavító kódok.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- Titkosírás (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.
  Például az RSA-módszer azt az elvet alkalmazza, hogy nagy számok nem bonthatók gyorsan prímek szorzatára.
- Forráskódolás: adatok tömörítése.
  Kevesebb tárolóhely, gyorsabb adattovábbítás.
- Hibajelző és hibajavító kódok.
  A megváltoztatott adatot zajos „csatornán” továbbítjuk.

A kódok típusai

Kódolás: adatok megváltoztatása.
Dekódolás: a megváltoztatott adatból az eredeti visszanyerése.

Célok

- Titkosírás (kriptográfia).
  A megváltoztatott adat illetéktelenek által nem olvasható.
  Például az RSA-módszer azt az elvet alkalmazza, hogy nagy számok nem bonthatók gyorsan prímek szorzatára.
- Forráskódolás: adatok tömörítése.
  Kevesebb tárolóhely, gyorsabb adattovábbítás.
- Hibajelző és hibajavító kódok.
  A megváltoztatott adatot zajos „csatornán” továbbítjuk.
  A címzett mégis képes lehet visszaállítani az eredetit.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).
- Mobiltelefonon át való kapcsolatteremtés.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).
- Mobiltelefonon át való kapcsolatteremtés.

A kódolás szempontjai

- Minél több hiba felismerhető/javítható legyen.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).
- Mobiltelefonon át való kapcsolatteremtés.

A kódolás szempontjai

- Minél több hiba felismerhető/javítható legyen.
- Mégis, minél kevésbé hosszabbodjon meg az üzenet.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).
- Mobiltelefonon át való kapcsolatteremtés.

A kódolás szempontjai

- Minél több hiba felismerhető/javítható legyen.
- Mégis, minél kevésbé hosszabbodjon meg az üzenet.
- Elegendően gyors kódolás/dekódolás.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).
- Mobiltelefonon át való kapcsolatteremtés.

A kódolás szempontjai

- Minél több hiba felismerhető/javítható legyen.
- Mégis, minél kevésbé hosszabbodjon meg az üzenet.
- Elegendően gyors kódolás/dekódolás.
- A csatorna tipikus hibáinak a jellege.

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).
- Mobiltelefonon át való kapcsolatteremtés.

A kódolás szempontjai

- Minél több hiba felismerhető/javítható legyen.
- Mégis, minél kevésbé hosszabbodjon meg az üzenet.
- Elegendően gyors kódolás/dekódolás.
- A csatorna tipikus hibáinak a jellege. Például betücsere;

A hibajavító kódok alkalmazási területei

- Adattárolás merevlemezen, kompakt lemezen, melynek egyes részei meghibásodhatnak.
- Egy űrszonda elküldi a képeket, mérési adatokat.
- Televíziós műsorszórás (például műholdról).
- Mobiltelefonon át való kapcsolatteremtés.

A kódolás szempontjai

- Minél több hiba felismerhető/javítható legyen.
- Mégis, minél kevésbé hosszabbodjon meg az üzenet.
- Elegendően gyors kódolás/dekódolás.
- A csatorna tipikus hibáinak a jellege.
  Például betücsere; sok egymás melletti betű hibája.

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 0

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 0 kódolva 000

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 00 kódolva 000

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 00 kódolva 000000

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001 kódolva 000000

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001 kódolva 000000111

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 0010 kódolva 000000111

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 0010 kódolva 000000111000

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 00101 kódolva 000000111000

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 00101 kódolva 000000111000111

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon:

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: aaa, aab, aba, $baa \mapsto a$.

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: aaa, aab, aba, $baa \mapsto a$.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás,

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni (karcolás az adathordozón,

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni (karcolás az adathordozón, sercenés rádióvételkor,

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni (karcolás az adathordozón, sercenés rádióvételkor, neve csomós hiba,

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni (karcolás az adathordozón, sercenés rádióvételkor, neve csomós hiba, angolul burst error),

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni (karcolás az adathordozón, sercenés rádióvételkor, neve csomós hiba, angolul burst error), akkor érdemes a betűket még össze is keverni

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni (karcolás az adathordozón, sercenés rádióvételkor, neve csomós hiba, angolul burst error), akkor érdemes a betűket még össze is keverni (kódátfűzés).

Háromszorozás

9.1.2. Példa

Minden betűt háromszor egymás után elküldünk.
Például 001011 kódolva 000000111000111111.
Dekódolás: többségi alapon: *aaa*, *aab*, *aba*, *baa* $\mapsto$ *a*.

Elemzés

Ha bármely három szomszédos betűből legfeljebb egy hibás, akkor az eredeti üzenet visszakapható (1-hibajavító kód).
Az üzenet háromszorosára nyúlik.
Ha szomszédos bitek hajlamosak egyszerre meghibásodni (karcolás az adathordozón, sercenés rádióvételkor, neve csomós hiba, angolul burst error), akkor érdemes a betűket még össze is keverni (kódátfűzés).
Ha betű kimaradhat, akkor szinkronjelek is kellhetnek.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Példa

A háromszorozásnál $Q = \{0, 1\}$,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Példa

A háromszorozásnál $Q = \{0, 1\}$, $k = 1$,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Példa

A háromszorozásnál $Q = \{0, 1\}$, $k = 1$, $n = 3$,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Példa

A háromszorozásnál $Q = \{0, 1\}$, $k = 1$, $n = 3$, $\varphi(x) = xxx$,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Példa

A háromszorozásnál $Q = \{0, 1\}$, $k = 1$, $n = 3$, $\varphi(x) = xxx$,
$C = \{000, 111\} \subseteq Q^3$.

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Példa

A háromszorozásnál $Q = \{0, 1\}$, $k = 1$, $n = 3$, $\varphi(x) = xxx$,
$C = \{000, 111\} \subseteq Q^3$. Ez egy 3 hosszú,

Alapfogalmak, jelölések

9.1.1. Definíció

A betűk halmaza Q, ez az ábécé, elemszáma q.
A Q elemeiből készített k hosszú sorozatok: szavak.
Halmazuk Q^k (a betűket vessző nélkül egymás mellé írjuk).
Kódolás: $\varphi : Q^k \to Q^n$ injektív függvény.
$\varphi(Q^k) = C \subseteq Q^n$ a φ értékkészlete, elemei a kódszavak.
A $C \subseteq Q^n$ egy (n, k) paraméterű kód. Az n a kód hossza.

A kód megadásakor sokszor csak a C halmaz szerepel,
φ kódoló függvény nem.

Példa

A háromszorozásnál $Q = \{0, 1\}$, $k = 1$, $n = 3$, $\varphi(x) = xxx$,
$C = \{000, 111\} \subseteq Q^3$. Ez egy 3 hosszú, $(3, 1)$ paraméterű kód.

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító,

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító, ha bárhogy veszünk két $v \neq w$ kódszót,
ha v-t is és w-t is legfeljebb t helyen megváltoztatjuk

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító, ha bárhogy veszünk két $v \neq w$ kódszót,
ha v-t is és w-t is legfeljebb t helyen megváltoztatjuk
(ezek a helyek mások lehetnek v, mint w esetében),

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító, ha bárhogy veszünk két $v \neq w$ kódszót,
ha v-t is és w-t is legfeljebb t helyen megváltoztatjuk
(ezek a helyek mások lehetnek v, mint w esetében),
akkor nem kaphatjuk Q^n-nek ugyanazt az elemét.

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító, ha bárhogy veszünk két $v \neq w$ kódszót,
ha v-t is és w-t is legfeljebb t helyen megváltoztatjuk
(ezek a helyek mások lehetnek v, mint w esetében),
akkor nem kaphatjuk Q^n-nek ugyanazt az elemét.

Például a háromszorozó kód 1-hibajavító

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító, ha bárhogy veszünk két $v \neq w$ kódszót,
ha v-t is és w-t is legfeljebb t helyen megváltoztatjuk
(ezek a helyek mások lehetnek v, mint w esetében),
akkor nem kaphatjuk Q^n-nek ugyanazt az elemét.

Például a háromszorozó kód 1-hibajavító és 2-hibajelző.

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító, ha bárhogy veszünk két $v \neq w$ kódszót,
ha v-t is és w-t is legfeljebb t helyen megváltoztatjuk
(ezek a helyek mások lehetnek v, mint w esetében),
akkor nem kaphatjuk Q^n-nek ugyanazt az elemét.

Például a háromszorozó kód 1-hibajavító és 2-hibajelző.
Ha 2 helyen megváltozik 000, akkor az nem kódszó.

Hibajelzés, hibajavítás

Érkezik egy $u \in Q^n$, ami nem kódszó (és így hiba történt).
Keresünk egy kódszót, amitől u a legkevesebb helyen tér el.

9.1.3. Definíció

Legyen $t \geq 1$ egész szám. A $C \subseteq Q^n$ kód t-hibajelző,
ha egy kódszót legfeljebb t helyen megváltoztatva
az eredmény nem lehet kódszó.
A C kód t-hibajavító, ha bárhogy veszünk két $v \neq w$ kódszót,
ha v-t is és w-t is legfeljebb t helyen megváltoztatjuk
(ezek a helyek mások lehetnek v, mint w esetében),
akkor nem kaphatjuk Q^n-nek ugyanazt az elemét.

Például a háromszorozó kód 1-hibajavító és 2-hibajelző.
Ha 2 helyen megváltozik 000, akkor az nem kódszó.
Ha 1 helyen változik, akkor rekonstruálható az eredeti.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Például $d(0010110010, 0110111000) = 3$.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Például $d(0010110010, 0110111000) = 3$.

A $C \subseteq Q^n$ kód minimális távolsága a különböző kódszavak Hamming-távolságainak minimuma.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Például $d(0010110010, 0110111000) = 3$.

A $C \subseteq Q^n$ kód minimális távolsága a különböző kódszavak Hamming-távolságainak minimuma. Jele: $d(C)$.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Például $d(0010110010, 0110111000) = 3$.

A $C \subseteq Q^n$ kód minimális távolsága a különböző kódszavak Hamming-távolságainak minimuma. Jele: $d(C)$.
Vagyis két legközelebbi kódszó távolsága.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Például $d(0010110010, 0110111000) = 3$.

A $C \subseteq Q^n$ kód minimális távolsága a különböző kódszavak Hamming-távolságainak minimuma. Jele: $d(C)$.
Vagyis két legközelebbi kódszó távolsága.

Például a háromszorozó kód minimális távolsága 3.

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Például $d(0010110010, 0110111000) = 3$.

A $C \subseteq Q^n$ kód minimális távolsága a különböző kódszavak Hamming-távolságainak minimuma. Jele: $d(C)$.
Vagyis két legközelebbi kódszó távolsága.

Például a háromszorozó kód minimális távolsága 3.

9.1.6. Gyakorlat (HF)

A C kód pontosan akkor t-hibajelző, ha $t < d(C)$,

Hamming-távolság

9.1.4. Definíció

A $v, w \in Q^n$ Hamming-távolsága azoknak a koordinátáknak a száma, ahol a két szó eltér. Jele: $d(v, w)$.

Például $d(0010110010, 0110111000) = 3$.

A $C \subseteq Q^n$ kód minimális távolsága a különböző kódszavak Hamming-távolságainak minimuma. Jele: $d(C)$.
Vagyis két legközelebbi kódszó távolsága.

Például a háromszorozó kód minimális távolsága 3.

9.1.6. Gyakorlat (HF)

A C kód pontosan akkor t-hibajelző, ha $t < d(C)$,
és pontosan akkor t-hibajavító, ha $2t < d(C)$.

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú,

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson,

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.
- Minél kevésbé nőjön az üzenet,

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.
- Minél kevésbé nőjön az üzenet, azaz $n - k$ kicsi legyen.

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.
- Minél kevésbé nőjön az üzenet, azaz $n - k$ kicsi legyen.

9.1.7. Hamming-korlát, NB

Ha $2t < d$, akkor $q^{n-k} = \frac{q^n}{|C|} \geq \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$.

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.
- Minél kevésbé nőjön az üzenet, azaz $n - k$ kicsi legyen.

9.1.7. Hamming-korlát, NB

Ha $2t < d$, akkor $q^{n-k} = \frac{q^n}{|C|} \geq \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$.

Perfekt kód: egyenlőség áll,

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.
- Minél kevésbé nőjön az üzenet, azaz $n - k$ kicsi legyen.

9.1.7. Hamming-korlát, NB

Ha $2t < d$, akkor $q^{n-k} = \frac{q^n}{|C|} \geq \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$.

Perfekt kód: egyenlőség áll, azaz minden $u \in Q^n$ szóhoz van tőle legfeljebb t Hamming-távolságra eső kódszó.

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.
- Minél kevésbé nőjön az üzenet, azaz $n - k$ kicsi legyen.

9.1.7. Hamming-korlát, NB

Ha $2t < d$, akkor $q^{n-k} = \dfrac{q^n}{|C|} \geq \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$.

Perfekt kód: egyenlőség áll, azaz minden $u \in Q^n$ szóhoz van tőle legfeljebb t Hamming-távolságra eső kódszó.

9.1.9. Singleton-korlát, NB

$$q^{n-k} = \frac{q^n}{|C|} \geq q^{d-1},$$

Korlátok

$C \subseteq Q^n$ egy d minimális távolságú, (n, k) paraméterű kód.
Ellentmondó követelmények:

- Minél több hibát javítson, azaz d nagy legyen.
- Minél kevésbé nőjön az üzenet, azaz $n - k$ kicsi legyen.

9.1.7. Hamming-korlát, NB

Ha $2t < d$, akkor $q^{n-k} = \frac{q^n}{|C|} \geq \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$.

Perfekt kód: egyenlőség áll, azaz minden $u \in Q^n$ szóhoz van tőle legfeljebb t Hamming-távolságra eső kódszó.

9.1.9. Singleton-korlát, NB

$q^{n-k} = \frac{q^n}{|C|} \geq q^{d-1}$, azaz $n - k \geq d - 1$.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek,

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú. $C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód,

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód, a kódolás a generátorpolinommal való szorzás.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód, a kódolás a generátorpolinommal való szorzás.

Példa: $Q = \{0, 1\}$ a kételemű test

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód, a kódolás a generátorpolinommal való szorzás.

Példa: $Q = \{0, 1\}$ a kételemű test és $g(x) = x^2 + x + 1$.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód, a kódolás a generátorpolinommal való szorzás.

Példa: $Q = \{0, 1\}$ a kételemű test és $g(x) = x^2 + x + 1$.
Ekkor $k = 1$, $n = 3$ esetén a háromszorozó kódolást kapjuk.

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód, a kódolás a generátorpolinommal való szorzás.

Példa: $Q = \{0, 1\}$ a kételemű test és $g(x) = x^2 + x + 1$.
Ekkor $k = 1$, $n = 3$ esetén a háromszorozó kódolást kapjuk.
Akkor is ha Q tetszőleges test:

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód, a kódolás a generátorpolinommal való szorzás.

Példa: $Q = \{0, 1\}$ a kételemű test és $g(x) = x^2 + x + 1$.
Ekkor $k = 1$, $n = 3$ esetén a háromszorozó kódolást kapjuk.
Akkor is ha Q tetszőleges test: ha $u \in Q$ konstans polinom,

Lineáris kód

9.2.1. Definíció

Ha Q egy véges test és C altere a Q^n vektortérnek, akkor C lineáris kód.

Ilyenkor $d(u, v) = d(u - v, 0)$. A $d(w, 0)$ a w súlya.

9.3.1. Definíció

Az $u = u_1 u_2 \dots u_k$ szó helyett az $u_1 x^{k-1} + \dots + u_{k-1} x + u_k$ polinomot tekintjük. Legyen $g \in Q[x]$ rögzített, $n - k$ fokú.
$C = \{g(x)u(x) \ : \ \mathrm{gr}(u) \leq k\}$ a g generátorú polinomkód.
Ez lineáris kód, a kódolás a generátorpolinommal való szorzás.

Példa: $Q = \{0, 1\}$ a kételemű test és $g(x) = x^2 + x + 1$.
Ekkor $k = 1$, $n = 3$ esetén a háromszorozó kódolást kapjuk.
Akkor is ha Q tetszőleges test: ha $u \in Q$ konstans polinom, akkor $g(x)u = ux^2 + ux + u \leftrightarrow uuu$.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített,

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű,

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$,

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$,

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

9.3.3. Állítás

A fenti Reed–Solomon-kód minimális távolsága pontosan d.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

9.3.3. Állítás

A fenti Reed–Solomon-kód minimális távolsága pontosan d.

Ez közel optimális: a Singleton-korlátban egyenlőség van.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

9.3.3. Állítás

A fenti Reed–Solomon-kód minimális távolsága pontosan d.

Ez közel optimális: a Singleton-korlátban egyenlőség van.
BCH-kód

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

9.3.3. Állítás

A fenti Reed–Solomon-kód minimális távolsága pontosan d.

Ez közel optimális: a Singleton-korlátban egyenlőség van.
BCH-kód (Bose, Ray-Chaudhuri, Hocquenghem):

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

9.3.3. Állítás

A fenti Reed–Solomon-kód minimális távolsága pontosan d.

Ez közel optimális: a Singleton-korlátban egyenlőség van.
BCH-kód (Bose, Ray-Chaudhuri, Hocquenghem):
az előző általánosítása, amikor α a Q egy bővítésében van,

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

9.3.3. Állítás

A fenti Reed–Solomon-kód minimális távolsága pontosan d.

Ez közel optimális: a Singleton-korlátban egyenlőség van.
BCH-kód (Bose, Ray-Chaudhuri, Hocquenghem):
az előző általánosítása, amikor α a Q egy bővítésében van,
g pedig $m_\alpha, \dots, m_{\alpha^{d-1}}$ legkisebb közös többszöröse.

Reed–Solomon-kód

9.3.5. Definíció

Legyen $0 \neq \alpha \in Q$ rögzített, a szorzásra legalább n rendű, $d \leq n$, és $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
Reed–Solomon-kód: a g generátorpolinomú, n hosszú kód.
A d szám a kód tervezett távolsága.

Ekkor $n - k = \mathrm{gr}(g) = d - 1$, vagyis $k = n - d + 1$.

9.3.3. Állítás

A fenti Reed–Solomon-kód minimális távolsága pontosan d.

Ez közel optimális: a Singleton-korlátban egyenlőség van.
BCH-kód (Bose, Ray-Chaudhuri, Hocquenghem):
az előző általánosítása, amikor α a Q egy bővítésében van,
g pedig $m_\alpha, \dots, m_{\alpha^{d-1}}$ legkisebb közös többszöröse.
(Jobb, mert kevesebb a betű.)

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re
$g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re $g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re $g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.
A $g(x)u(x)$-nek gyöke α^j, ha $1 \leq j \leq d - 1$.

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re
$g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.
A $g(x)u(x)$-nek gyöke α^j, ha $1 \leq j \leq d - 1$.
A $g(\alpha^j)u(\alpha^j) = 0$ egyenleteket $1 \leq j \leq m (\leq d - 1)$-re
tekintsük lineáris egyenletrendszernek $g(x)u(x)$ együtthatóira.

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re
$g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.
A $g(x)u(x)$-nek gyöke α^j, ha $1 \leq j \leq d - 1$.
A $g(\alpha^j)u(\alpha^j) = 0$ egyenleteket $1 \leq j \leq m (\leq d - 1)$-re
tekintsük lineáris egyenletrendszernek $g(x)u(x)$ együtthatóira.
Determinánsa „majdnem” Vandermonde-determináns,

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re
$g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.
A $g(x)u(x)$-nek gyöke α^j, ha $1 \leq j \leq d - 1$.
A $g(\alpha^j)u(\alpha^j) = 0$ egyenleteket $1 \leq j \leq m (\leq d - 1)$-re
tekintsük lineáris egyenletrendszernek $g(x)u(x)$ együtthatóira.
Determinánsa „majdnem” Vandermonde-determináns,
ami az $o(\alpha) \geq n > \mathrm{gr}(gu)$ feltétel miatt nem nulla

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re
$g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.
A $g(x)u(x)$-nek gyöke α^j, ha $1 \leq j \leq d - 1$.
A $g(\alpha^j)u(\alpha^j) = 0$ egyenleteket $1 \leq j \leq m (\leq d - 1)$-re
tekintsük lineáris egyenletrendszernek $g(x)u(x)$ együtthatóira.
Determinánsa „majdnem” Vandermonde-determináns,
ami az $o(\alpha) \geq n > \mathrm{gr}(gu)$ feltétel miatt nem nulla (HF).

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re
$g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.
A $g(x)u(x)$-nek gyöke α^j, ha $1 \leq j \leq d - 1$.
A $g(\alpha^j)u(\alpha^j) = 0$ egyenleteket $1 \leq j \leq m (\leq d - 1)$-re
tekintsük lineáris egyenletrendszernek $g(x)u(x)$ együtthatóira.
Determinánsa „majdnem” Vandermonde-determináns,
ami az $o(\alpha) \geq n > \mathrm{gr}(gu)$ feltétel miatt nem nulla (HF).
Ezért ennek a homogén lineáris egyenletrendszernek
csak triviális megoldása van:

Reed–Solomon-kód: bizonyítás

9.3.3. Bizonyítás

Belátjuk, hogy a Reed–Solomon-kód minimális távolsága d.
Tudjuk, hogy $g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})$.
A $g(x) \cdot 1$ kódszó súlya legfeljebb $d = \mathrm{gr}(g) + 1$.
Megfordítva: Be kell látni, hogy minden $0 \neq u(x)$-re
$g(x)u(x)$-nek legalább d darab nem nulla együtthatója van.
Indirekt feltevés: $m < d$ darab nem nulla együttható van.
A $g(x)u(x)$-nek gyöke α^j, ha $1 \leq j \leq d - 1$.
A $g(\alpha^j)u(\alpha^j) = 0$ egyenleteket $1 \leq j \leq m (\leq d - 1)$-re
tekintsük lineáris egyenletrendszernek $g(x)u(x)$ együtthatóira.
Determinánsa „majdnem” Vandermonde-determináns,
ami az $o(\alpha) \geq n > \mathrm{gr}(gu)$ feltétel miatt nem nulla (HF).
Ezért ennek a homogén lineáris egyenletrendszernek
csak triviális megoldása van: az, amikor $g(x)u(x) = 0$. □

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$,

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$,

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$,

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$,

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD-ken egymás után kétféle kódot is használnak:

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD-ken egymás után kétféle kódot is használnak:
az n értéke egyszer 28,

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD-ken egymás után kétféle kódot is használnak:
az n értéke egyszer 28, egyszer 32.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD-ken egymás után kétféle kódot is használnak:
az n értéke egyszer 28, egyszer 32.
Mindkétszer a kódátfűzés módszerével is ötvözik.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD-ken egymás után kétféle kódot is használnak:
az n értéke egyszer 28, egyszer 32.
Mindkétszer a kódátfűzés módszerével is ötvözik.
A végén $8 \leftrightarrow 14$ bites átalakító táblázat.

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD-ken egymás után kétféle kódot is használnak:
az n értéke egyszer 28, egyszer 32.
Mindkétszer a kódátfűzés módszerével is ötvözik.
A végén $8 \leftrightarrow 14$ bites átalakító táblázat.
Jobb lejátszónak jobb dekódere lehet

A CD matematikája

9.5. Szakasz

Legyen $m(x) = x^8 + x^4 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$, ez irreducibilis.
Ezért $Q = \mathbb{F}_{256} \cong \mathbb{Z}_2[x]/(m)$, legyen $\alpha \in Q$ gyöke m-nek.
Ekkor minden betű egy byte, azaz egy 8-bites szó.
Belátható, hogy α rendje a szorzásra 255 (generálja $Q^\times$-et).
A tervezett távolság $d = 5$, hogy 2-hibajavító kódot kapjunk.
Azaz $n - k = 4$, és az α-hoz tartozó Reed-Solomon-kódban $n \leq o(\alpha) = 255$ lehetséges.

A CD-ken egymás után kétféle kódot is használnak:
az n értéke egyszer 28, egyszer 32.
Mindkétszer a kódátfűzés módszerével is ötvözik.
A végén $8 \leftrightarrow 14$ bites átalakító táblázat.
Jobb lejátszónak jobb dekódere lehet (több hibát javít).