Egzamino klausimai (rūšys)

1. **Bendrieji kombinatorikos dėsniai**

* **Sumavimo dėsnis**

Jei kokiam nors objektui A pasirinkti yra *m* būdų, o kitam objektui B pasirinkti yra *n* būdų, tai pasirinkti “arba A, arba B” yra *m + n* būdų

***Pastaba***.

Taikant taip suformuluotą sumavimo dėsnį, reikia žiūrėti, kad nei vienas objekto *A* pasirinkimo būdas nesutaptų su kuriuo nors objekto *B* pasirinkimo būdu. Jei tokių sutapimų yra *k*, tai susidaro *m* + *n* – *k* pasirinkimo būdų.

* **Sandaugos dėsnis**

Jei objektui A pasirinkti yra *m* būdų, o po kiekvieno tokio pasirinkimo objektą B galima pasirinkti *n* būdais, tai porą (A, B) pasirinkti nurodytąja tvarka galima *m ⋅ n* būdais.

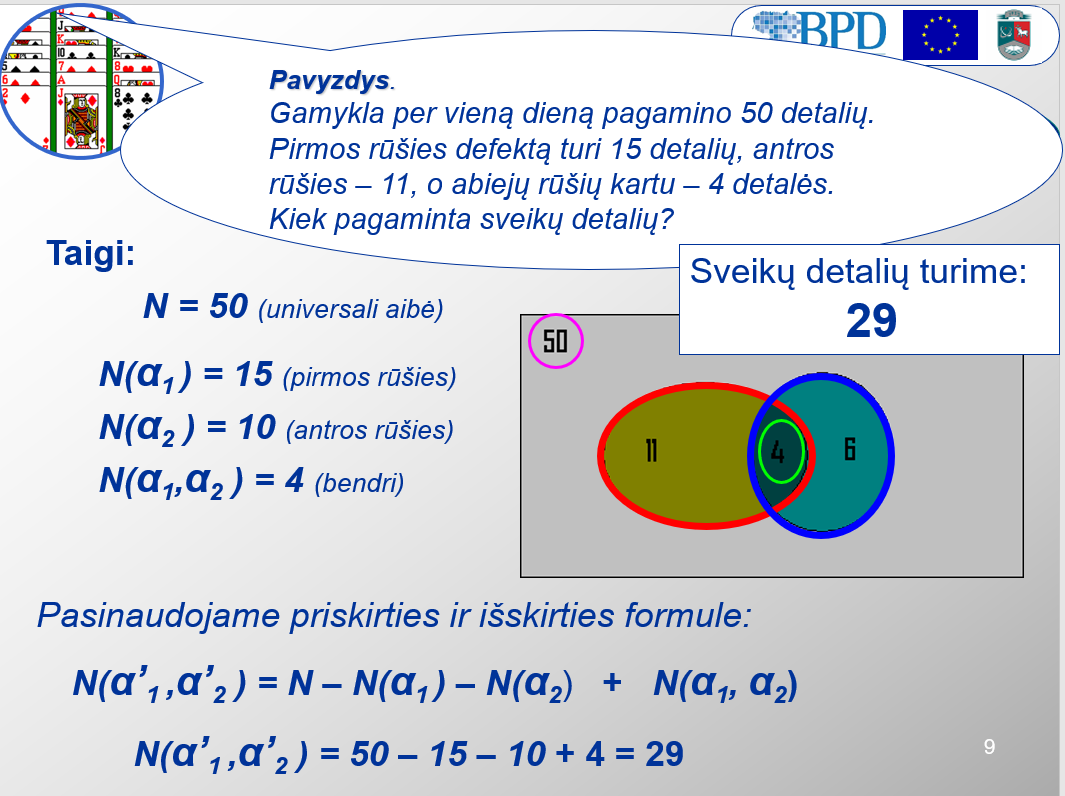
* **Priskirties ir išskirties dėsnis**

Sakykime, kad kai kuriems iš N turimų daiktų būdingos savybės. Simboliu N(αi , αj ,…, αk ) pažymėkime skaičių daiktų, turinčių savybes (į kitas tų daiktų savybes nekreipiame dėmesio). Jei reikės pabrėžti, kad imami tik tie daiktai, kurie neturi kurios nors savybės, tai tą savybę rašysime su brūkšneliu.

Pavyzdžiui, N(α1, α2 ,α’4 ) žymi skaičių daiktų, turinčių savybes α1 ir α2 , bet neturinčių savybės α’4 (į kitas jų savybes nekreipiame dėmesio).

***Tada priskirties ir išskirties dėsnis bus užrašomas taip:***

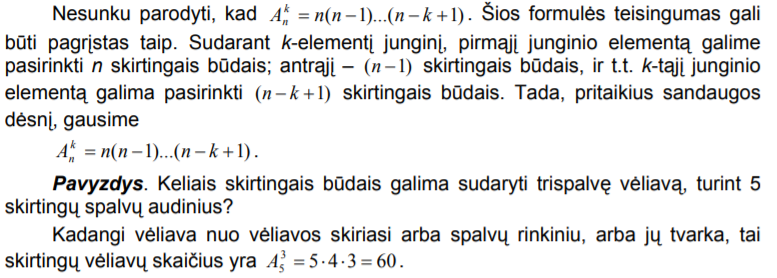
****



1. **Junginiai**
   1. **Pagrįsti junginių skaičiaus apskaičiavimą.**

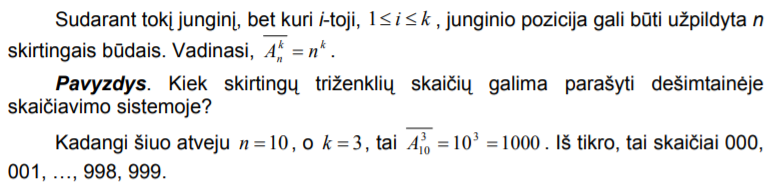
**Gretiniai be pasikartojimų**

Turime n skirtingų daiktų. Kiek iš jų galima sudaryti k-elemenčių junginių, sudarytų iš skirtingų elementų, kai junginys nuo junginio skiriasi arba bent vienu elementu, arba jų tvarka? Tokie junginiai vadinami gretiniais be pasikartojimų.



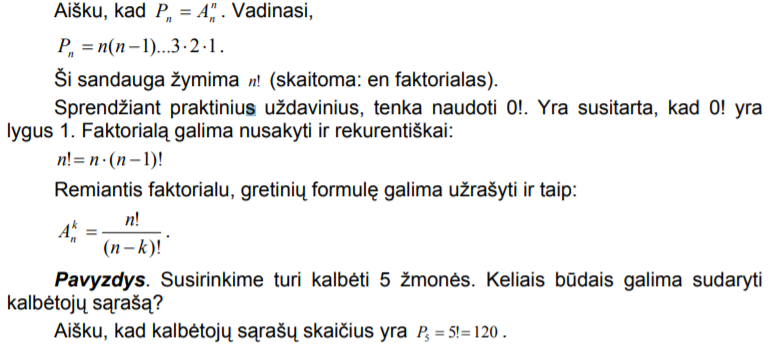
**Gretiniai su pasikartojimais**

Turime n skirtingų rūšių daiktų. Kiek iš jų galima sudaryti k-elemenčių junginių, sudarytų iš skirtingų elementų, kai junginys nuo junginio skiriasi arba bent vienu elementu, arba jų tvarka ir elementai junginyje gali kartotis? Tokie junginiai vadinami gretiniais su pasikartojančiais elementais.



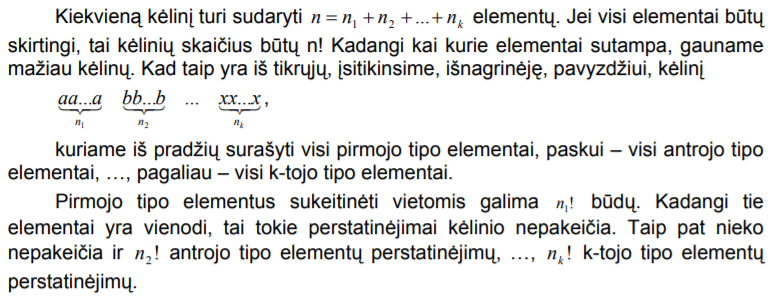
**Kėliniai be pasikartojimų**

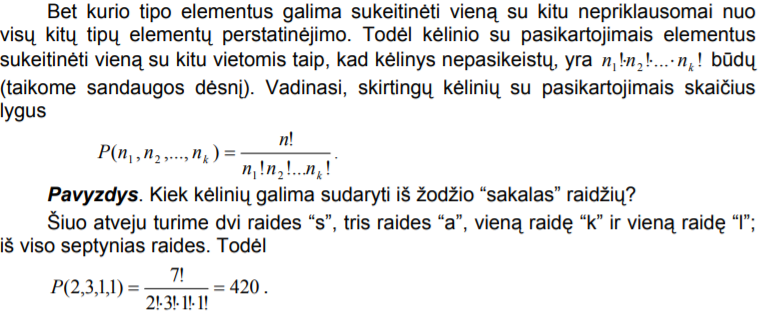
Sudarydami gretinius be pasikartojimų iš n elementų po k, gavome junginius, kurie vienas nuo kito skiriasi ir pačiais elementais, ir jų išsidėstymo tvarka. Tačiau, jei sudarinėtume junginius iš visų n elementų, tai jie galėtų skirtis vienas nuo kito tik juose esančių elementų tvarka. Tokie junginiai vadinami kėliniais iš n elementų arba, trumpiau, nelemenčiais kėliniais.



**Kėliniai su pasikartojančiais elementais**

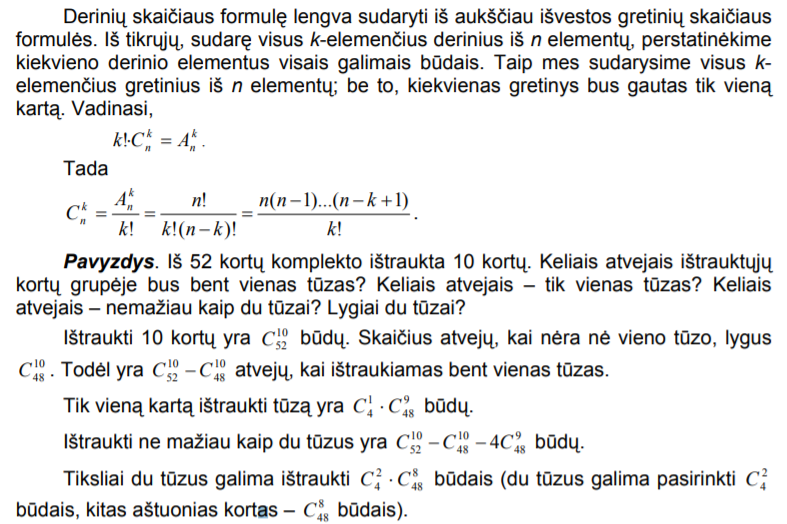
Turime k skirtingų tipų daiktus. Kiek kėlinų galima sudaryti iš n1 pirmojo tipo elementų, iš n2 antrojo tipo elementų, …,nk k-tojo tipo elementų?





**Deriniai be pasikartojimų**

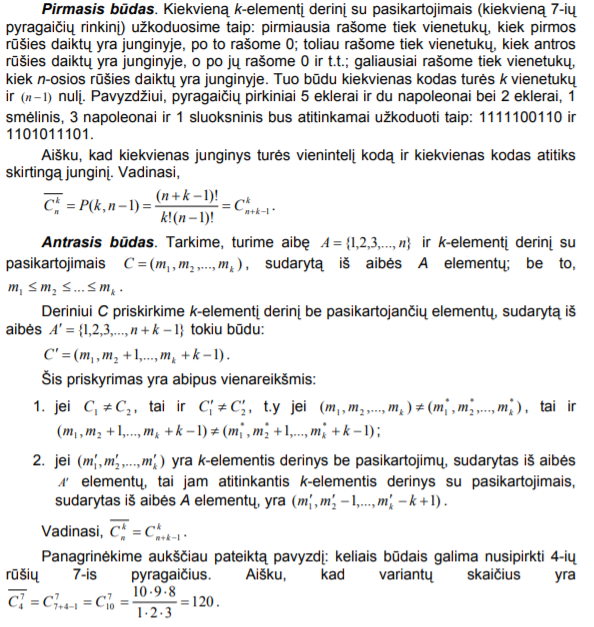
Visi galimi k-elemenčiai junginiai iš n elementų, kai junginys nuo junginio skiriasi bent vienu elementu, vadinami deriniais.



**Deriniai su pasikartojimais**

Turime n skirtingų rūšių daiktų. Kiek skirtingų k-elemenčių junginių iš n skirtingų rūšių daiktų galima sudaryti, jei junginys nuo junginio skiriasi bent vienu elementu ir elementai junginyje gali kartotis?

Pavyzdžiui, konditerijos parduotuvėje yra 4 rūšių pyragaičių: eklerų, smėlinių, napoleonų ir sluoksninių. Keliais būdais galima nusipirkti 7 pyragaičius?



* 1. **Apskaičiuoti junginių skaičių arba tikimybę.**

1. **Junginių generavimas**
   1. **Bendra junginių generavimo schema.**

Junginys = SugeneruotiPradiniJungini();

IteracijosNr = 1;

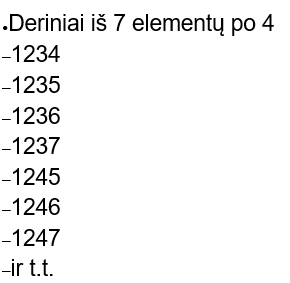
while YraTolesniuJunginiu(Junginys, IteracijosNr)

Junginys = SugeneruotiTolesniJungini(Junginys, IteracijosNr);

IteracijosNr = IteracijosNr + 1;

end

* 1. **Derinių generavimas leksikografine tvarka.**
* Pirmasis (mažiausias) derinys gali būti sugeneruotas tiesiog išvardijant pirmuosius k skaičių didėjimo tvarka (1,2.., k)
* Kiekvienameingsnyje norėsime sugeneruoti vis didesnį n-etainį skaičių atitinkantį derinį ir, aišku, nei vieno nepraleisti.
* Turime rasti pirmą elementą nuo galo, kurį galime padidinti.
* Jis bus mažesnis už atitinkamą didžiausio derinio elementą.
* Tą elementą padidiname
* Už jo einančius elementus surašome kuo mažesnius
* Kadangi jie turi būti didesni už prieš tai einančius, rašysime vis vienetu didesnius.



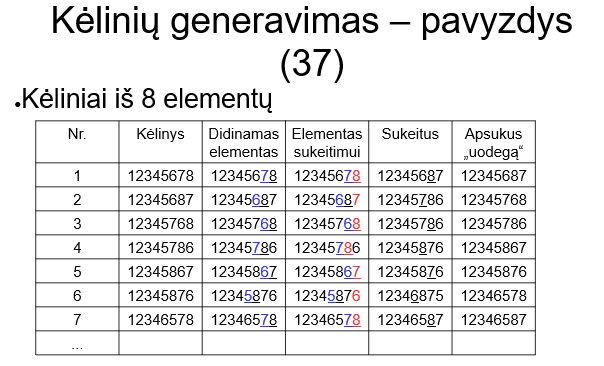
* 1. **Kėlinių generavimas leksikografine tvarka.**

Pasiruošimas

* Kiekvieną objektą galima pažymėti skaičiumi
* Kėlinius galima žymėti šių skaičių sąrašais
* Skaičius išdėstysime nebe būtinai didėjimo tvarka
* Kėliniuose elementų tvarka svarbi
* Skaičių sąrašai atitiks n-etainės skaičiavimo sistemos skaičius
* Mažiausias kėlinys iš n elementų bus <1,2 ..., n>
* Didžiausias kėlinys bus <n, ..., 2, 1>

Generavimas

* Pirmiausias (mažiausias) kėlinys gali būti sugeneruotas tiesiog išvardinant pirmuosius n skaičių didėjimo tvarka (1, 2, ..., n)
* Kiekviename žingsnyje mes norėsime sugeneruoti vis didesnį n-etainį skaičių, atitinkantį kėlinį ir nei vieno nepraleisti.
* Ieškosime pirmo elemento nuo galo, kurį galėtumėme padidinti.
* Padidinti galėsime sukeisdami su kitu, arčiau galo esančiu skaičiumi.
* Tas skaičius turės būti didesnis
* Lyginti užteks gretimas skaičių poras.
* Radę tinkamą porą, turėsime rasti skaičių, kurį įrašę vietoj „didinamojo“, gausime kėlinį, kuris būtų kuo mažesnis, bet didesnis už esamą.
* Tad reikės rasti kuo mažesnį skaičių, kuris yra didesnis už „didinamąjį“ ir eina už jo
* Galima pasinaudoti tuo, kad elementai už „didinamojo“ yra išsidėstę mažėjimo tvarka
* Kitaip vienas iš jų pats būtų „didinamasis“
* Radę „tinkamą“ elementą, jį galime sukeisti vietomis su „didinamuoju“
* Už jo reikia surašyti likusius elementus taip, kad kėlinys būtų kuo mažesnis
  + Juos reikės išdėstyti didėjimo tvarka
  + O tai galima padaryti surašius juos atvirkščiai
  + Nes jie išsidėstę mažėjimo tvarka



* 1. **Sugeneruoti junginį ar keletą junginių.**

1. **Rekurentiniai sąryšiai:**
   1. **Rekurentinio sąryšio sąvoka.**

Rekurentinis sąryšis nusako seką skaičių taip, kad tolesni sekos elementai apskaičiuojami iš prieš juos einančių

* 1. **Tiesiniai rekurentiniai sąryšiai su pastoviais koeficientais ir jų sprendimas.**

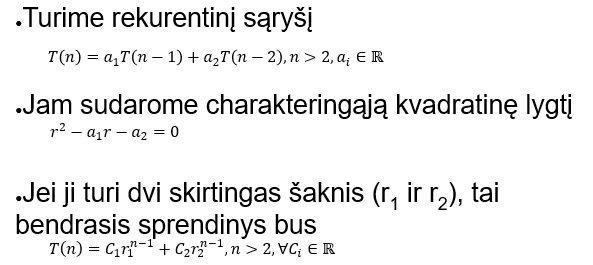
Tiesiniai rekurentiniai sąryšiai su pastoviais koeficientai nusakomi taip:

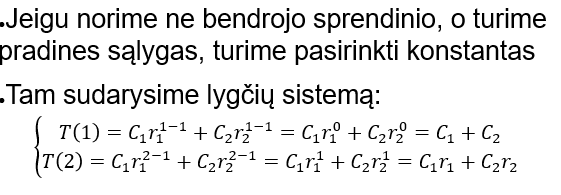


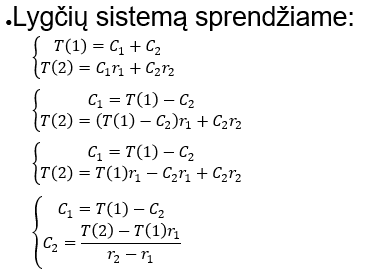
Kitaip tariant, tolesnis narys yra suma ankstesnių narių su realiaisias koeficientais.

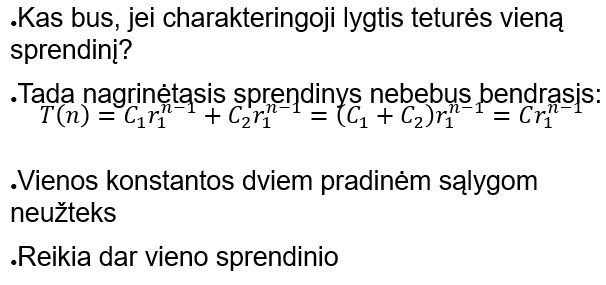
Kaip spręsti tokius uždavinius:

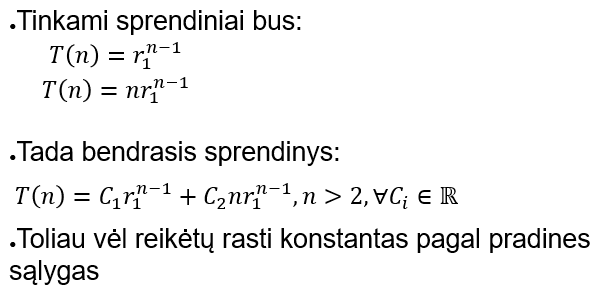












Jei turėsime ne antrojo, o aukštesnio laipsnio rekurentinį ryšį, jį spresime panašiai:

* Sudarysime charakteringąją lygtį
* Ją išspręsime
* Jei jos šaknys nesikartoja, dauginsime jas iš konstantų ir sudėsime
* Besikartojančias dar dauginsime iš n laipsnių
  1. **Rekurentinių sąryšių sudarymas.**
     1. **Sudaryti rekurentinį sąryšį duotam uždaviniui.**

1. **Kriptografija**
   1. **Kriptografija ir steganografija.**

Kriptografija – mokslas (ar matematikos šaka) apie informacijos perdavimą nesuteikiant jos „trečioms šalims“, neleidžiant pastarosioms jos pakeisti ir pan.

Kriptografijoje siekiama nuslėpti perduodamą informaciją, steganografijoje siekiama nuslėpti informacijos perdavimo faktą. Pabyzdžiui, pranešimas paslepiamas lyg kažkas kito.

* 1. **Kriptografinės atakos.**

Pagal tai, kas yra žinoma ar įmanoma kriptoanalitikui, skiriamos įvairios atakos:

* Šiftogramos ataka – yra žinoma tik šifrograma arba kelios šifrogramos
* Žinomos tekstogramos ataka – žinoma šifrograma ir ją atitinkanti tekstograma
* Parenkamos tesktogramos ataka -kriptoanalitikas gali parinkti tekstogramas ir gauti jas atitinkančias šifrogramas
* Parenkamos šifrogramos ataka – kriptoanalitikas gali gauti tekstogramas, kurios atitinka jo pasirinktas šifrogramas
* Susijusio rakto ataka – kriptoanalitikas gali stebėti veikimą su skirtingais, bet susijusiais raktais, pavyzdžiui tokiais, apie kuriuos žinoma, kad jie skiriasi vienu bitu.

\*žinoma galimi ir tarpiniai variantai ir pan.

* 1. **Atpažinti situacijoje kriptografinę ataką.**
  2. **Reikalavimai kriptosistemoms.**

Iš kriptosistemos galima norėti, kad ji užtikrintų:

* Konfidencialumą – tokiu atveju norima pranešti, kad pranešimo nebūtų galima skaityti trečiosioms šalims.
* Autentifikavimą – kad būtų galima nustatyti, kas siuntė pranešimą
* Intergralumą – kad trečios šalys negalėtų pakeisti pranešimo nepastebėtos
* Neatsisakymas – pranešimo siuntėjas negalėtų sukelti abejonių, ar jis yra siuntęs pranešimą

JAV armijos lauko statutas nurodo tokius reikalavimus idealiai karinei kriptosistemai:

* Suaigi nepriklausomai nuo perduodamų pranešimų kiekio
* Paprasta naudoti be ypatingo apmokymo
* Nesunku visus, kurie sistema naudoja, aprūpinti raktais
* Dirba visomis oro ir pan. Sąlygomis
* Nieko svarbaus neprarandama priešui perėmus sistemos „egzempliorių“

Taip pat paminima, kad tokių sistemų nėra, bet pastebėtina, kad saugumas nėra vienintelis reikalavimas 😊

1. **Vernamo šifras**
   1. **Kas tai yra? Kaip jis veikia?**

Vernamo šifras remiasi tuo, kad:

* Raktas yra tokio pat ilgio, kaip ir tekstograma
* Raktas susideda iš visiškai atsitiktinių simbolių
* 0 ir 1 tikimybė vienoda
* Kiekvienas bitas generuojamas nepriklausomai nuo kitų
* Raktas kiekvieną kartą generuojamas iš naujo
* Šifrograma gaunama sudedant raktą ir tekstogramą moduliu du:



* Laikoma, kad tekstograma, šifrograma ir raktas jau yra užkoduoti dvejetainiais skaičiais
  1. **Kodėl jis nenulaužiamas?**
* Bet kuri šifrograma gali būti gauta iš bet kokios tokio pat ilgio tekstogramos su kuriuo nors raktu
* Visi raktai yra vienodai tikėtini
* Tad šifrograma nesuteikia jokios papildomos informacijos apie tekstogramą

Net kai turime dalį tekstogramos, Vernamo šifras yra nenulaužiamas, nes

* Skirtingi rakto simboliai parenkami nepriklausomai
* Dėl to gausime tik tą dalį rakto, kuri bus nenaudinga

Net kai galime parinkti tekstogramą ir šifrogramą Vernamo šifras yra nenulaužiamas, nes:

* Kiekvieną kartą bus naudojamas vis kitas raktas
* Mes gausime tą raktą, kuris nebus pakartotas

Yra įrodyta, kad šifras yra nenulaužiamas tada ir tik tada, kai jis yra „labai panašus“ į Vernamo šifrą

* Raktas tokio pat ilgio, kai tekstograma
* Raktas visiškai atsitiktinis
* Raktas kas kart generuojamas iš naujo
  1. **Kokie jo trūkumai?**
* Vernamo šifras reikalauja visiškai atsitiktinio rakto
  + Pseudoatsitiktinių skaičių generatorius (kaip rand funkcijos kopmiuteriuose) netinka
* Vernamo šifras reikalauja iš anksto pasidalinti rinkiniu raktų, kurių yra bent tiek, kiek bus pranešimų, ir kurie yra bent tokio ilgio, kaip pranešimai
* Vernamo šifras – ne elektroninis parašas
  + Jeigu kažkaip pavyksta perimti ir tekstogramą ir šifrogramą, perduotą pranešimą galima pakeisti į bet kokį kitą tokio pat ilgio pranešimą.
* Raktą reikia užsirašyti, išmokti mintinai jo neverta ir nelabai įmanoma. O užrašas gali būti perimtas.
* Raktas negali būti pakartotas antrą kartą, nes tada šifras lengvai nulaužiamas

1. **Klasikiniai šifrai**
   1. **Pakeitimas ir perstatymas.**

Pakeitimas – vieni simboliai pakeičiami kitais.

Perstatymas – simboliai keičiami vietomis.

* 1. **Monoalfabetiniai pakeitimo šifrai.**

Vienas tekstogramos simbolis keičiamas į vieną šifrogramos simbolį

Tas pats simbolis visur keičiamas taip pat.

* + 1. **Cezario šifras.**

Cezario šifro atveju, kiekviena abėcėlės raidė pakeičiama į kitą, kuri abėcėlėje eina už kažkiek pozicijų. Šių pozicijų skaičius ir yra šifro raktas.

Cezario šifrą galima nulaužti išbandžius visus galimus variantus, pvz angliškoje abėcėlėje jų tėra 26.

* + 1. **Afininiai šifrai.**

Afininio šifro atveju kėlinys atkuriamas iš skaičių poros (a ir b). Skaičius a turi būti tarpusavyje pirminis su raidžių skaičiumi abėcėlėje (n)

Raidės užkoduojamos skaičiais (numeriais alfabete)

Tada užšifruojama funkcija:





* + 1. **Šifrų kriptoanalizė.**
    2. **Užšifruoti ar dešifruoti tekstą monoalfabetiniu pakeitimo šifru.**
    3. **Atlikti nesudėtingą kriptoanalizę.**

Apie situos nelabai yra ka pasakyti, arba supranti arba ne xD

* 1. **Monoalfabetiniai daugiaraidžiai pakeitimo šifrai.**

Naudojant tokius šifrus viena raidė tekstogramoje keičiama raidžių grupe šifrogramoje.

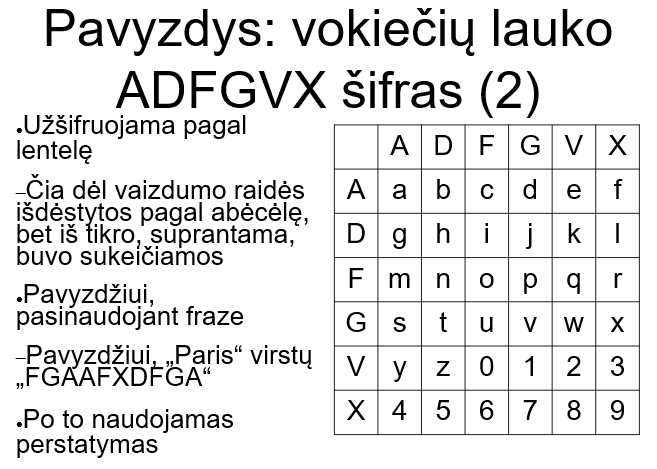
* + 1. **Pranašumai ir trūkumai.**

Pranašumai:

* Leidžia užšifruoti vieną raidę skirtingai, tuo paslepiant raidžių dažnių pasiskirtymą
* Užšifruoti skirybos ženklus, skaitmenis ir pan
* Toliau galima užšifruoti raidžių dvejetus, trejetus...

Trūkumai:

* Pranešimas pailgėja
* Šifravimas pasunkėja
  + 1. **Vokiečių lauko ADFGVX šifras.**



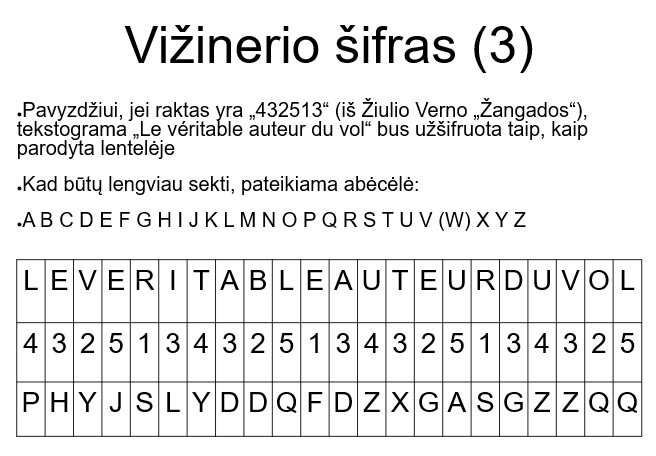
* 1. **Polialfabetiniai pakeitimo šifrai.**

Naudojant polialfabetinius pakeitimo šifrus vienas ir tas pats tekstogramos simbolis skirtingose vietose pakeičiamas skirtingais šifrogramos simboliais.

* + 1. **Vižinerio šifras.**

Šio šifro atveju, raktas yra frazė ar skaičių seka. Raktas periodiškai kartojamas tiek, kad kiekvienai tekstogramos raidei būtų priskirtas rakto skaičius ar raidė.

Tada kiekviena tekstogramos raidė užšifruojama lyg Cezario šifru, kaip raktą naudojant atitinkamą rakto skaičių ar raidę.



* + - 1. **Užšifruoti ar dešifruoti tekstą Vižinerio šifru.** Kaip nors lol
    1. **Polialfabetinio pakeitimo šifro kriptoanalizė.**
       1. **Nustatyti Vižinerio šifro rakto ilgį, kai duotos pasikartojančios raidžių sekos ir atstumai tarp jų.**

Dažniausiai pasikartojantis dažnių daliklis.

* 1. **Perstatymo šifrai.**

Perstatymo šifrai sukeičia tekstogramos raides vietomis.

Perstatymas lentele:

Vienas iš paprastų būdų tai padaryti – surašyti tekstogramą į stačiakampę lentelę, po to tekstas nuskaitomas kita tvarka.

Galima stulpelius dar sukeisti vietomims.

Perstatymas tinkleliu:

Tinklelis susideda iš langelių. Vieni leidžia rašyti per juos, kiti ne.

Užšifruojama taip:

* Tinklelis užpildomas tekstogramos raidėmis
* Tinklelis pasukamas stačiu kampu
* Tai kartojama, kol tinklelis bus pasuktas keturis kartus
* Tada tuščioje vietoje tas pats kartojama su likusia tekstogramos dalimi
* Šifro raktas yra tinklelis ir pusė, į kurią pasukama

1. **Dalumas, pirminiai skaičiai ir pan.**

Sakysime, kad skaičius a dalijasi iš skaičiaus b, jei a dalybos iš b liekana yra lygi 0. Tai žymima b|a (2|4| , 5|100, 15|30)

Dalumas gali būti apibrėžtas ir neigiamiems skaičiams.

Dalumo savybės:

* Jei du skaičiai dalijasi iš trečio, iš jo dalinsis ir jų suma.
* Jei du skaičiai dalijasi iš trečio, iš jo dalinsis ir jų skirtumas.
* Jei vienas skaičius dalijasi iš kito, iš pastarojo dalinsis ir pirmojo skaičiaus sandauga su bet kuriuo sveikuoju skaičiumi.

Dalikliai ir kartotiniai:

Skaičiaus 10 dalikliai yra 2 ir 5.

Skaičiaus 10 kartotiniai yra 20, 30... t.t.

Bendras didžiausias daliklis – didžiausias iš skaičių, iš kurių dalijasi duoti skaičiai.

Pirminiai skaičiai – tokie skaičiai, kurie dalijasi tik iš vieneto ir iš savęs. 1 NĖRA PIRMINIS SKAIČIUS

Tarpusavyje pirminiai skaičiai yra skačiai, kurių bendras daliklis yra tik vienetas (2 ir 3, 4 ir 9, 6 ir 35)

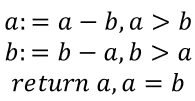
Pagrindinė aritmetikos teorema teigia, kad kiekvieną naturinį skaičių galima išskaidyti į pirminių daliklių sandaugą vienu ir tik vienu būdu.

Tai yra viena iš pagrindinių priežasčių, kodėl vienetas nėra priskiriamas prie pirminių skaičių.

Euklido teorema teigia, kad pirminių skaičių yra be galo daug.

1. **Euklido algoritmas.**

* Duoti du natūriniai skaičiai: a ir b
  + Iš didesnio skaičiaus atimame mažesnį
  + Skirtumą užrašome į didesnio vietą
  + Kartojame, kol vienas iš skaičių virs nuliu arba kol jie susilygins ir nebebus didesnio ir mažesnio
  + Taip gausime a ir b bendrą didžiausią daliklį



* 1. **Pritaikyti Euklido algoritmą.** Ez

1. **Išplėstinis Euklido algoritmas.**

Bezu tapatybė teigia, kad



Pastebėtina, kad x ir y yra ne iš natūrinių, o iš sveikųjų skaičių aibės – vienas iš jų paprastai bus neigiamas. Tokius x ir y (aišku ir BDD(a,b)) galima rasti išplėstiniu Euklido algoritmu.

* Iš pradžių turime du skaičius: a ir b
* Skaięiuosime xi, yi, di ir ki reikšmes taip, kad galiotų sąryšis:



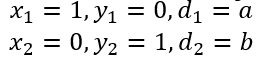
* Atlikdami išplėstus Euklido algoritmo žingsnius bandysime di vietoje gauti a ir b bendrą didžiausią daliklį.
* Tegu toliau a > b.
* Kaip iš pradžių patenkinti šį sąryšį?



* Žinome, kad:



* Tad inicializuojame:



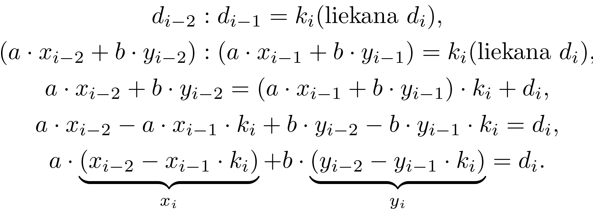
* Toliau pastebime, kad d1 ir d2 atitinka a ir b Euklido algoritme, tad vykdysime:



* Tada xi ir yi bus randami taip:



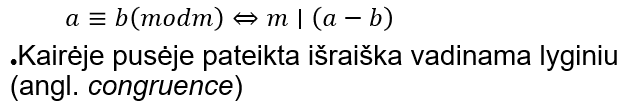
* Klausi kodėl? Įsitikinkime!



* 1. **Pritaikyti išplėstinį Euklido algoritmą.** Nu prie šito reiks bšk pasėdėt

1. **Modulinė aritmetika, lyginiai.**
   1. **Lyginio sąvoka.**

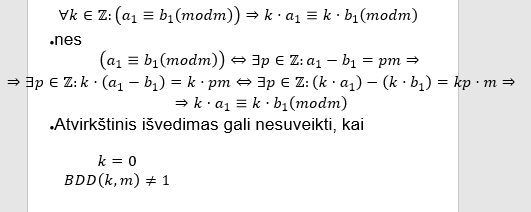
Skaičius a lygsta skaičiui b moduliu m, jei a ir b skirtumas yra m kartotinis.



* 1. **Moduliniai veiksmai.**

Sudėtis modulinėje artimetikoje veikia kaip suma moduliu du – iš esmės imama sumos dalybos iš modulio liekana.

Daugyba modulinėje aritmetikoje veikia panašiai



Jei kas suprantat sita skaidre parasykit man i fb aciu

* 1. **Moduliu atvirkštinis skaičius.**

Skaičius, moduliu atvirkštinis skaičiui a yra skaičius, kurio sandauga su duotoju skaičiumi duotu moduliu lygsta vienetui, panašiai kaip su paprastais skaičiais



Moduliu atvirkštinis skaičius egzistuoja tada ir tik tada, kai a ir m yra tarpusavyje pirminiai

Moduliu atvirkštinio skaičiaus radimas:

* Galime rasti pasinaudodami Euklido išplėstiniu algoritmu
* Iš esmės mes ieškome tokio x, su kuriuo



* Tada pagal lyginio apibrėžimą



* Vadinasi



* Perrašome



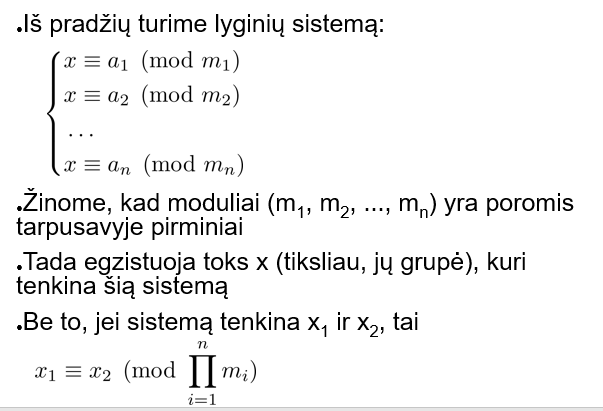
* Kitaip (perkeliame porą dėmenų į priešingą pusę ir keičiame vietomis dauginamuosius



* Gavome diofantinę lygtį, kurios forma yra

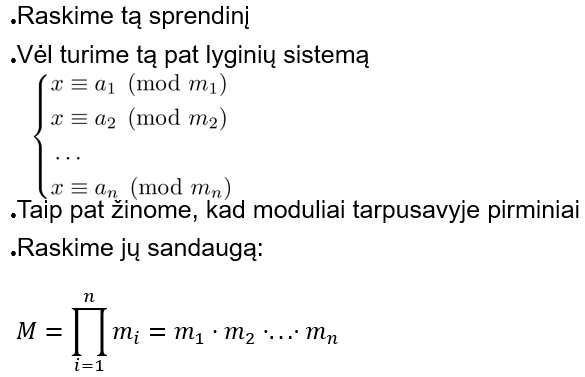


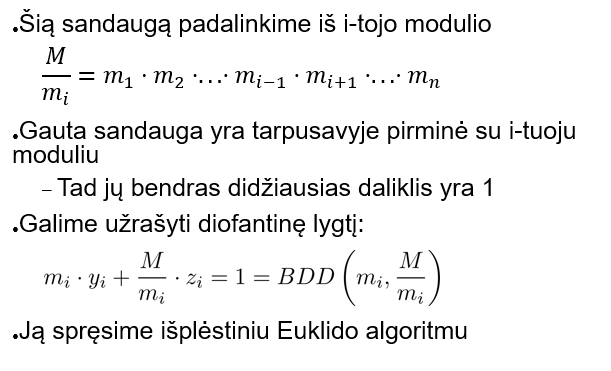
* O tokias lygtis sprendžiame Euklido algoritmu ir ieškome x ir q
  1. **Kinų teorema.** Dieve padėk

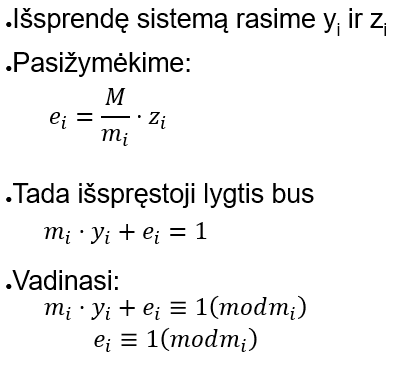


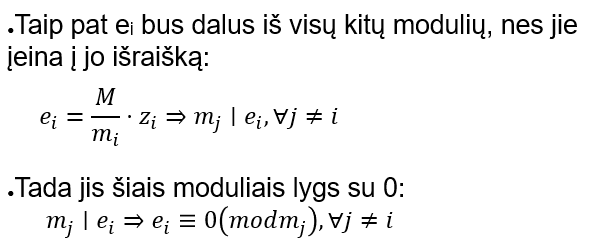
Wtf padėkit ir su šituo jei kas moka

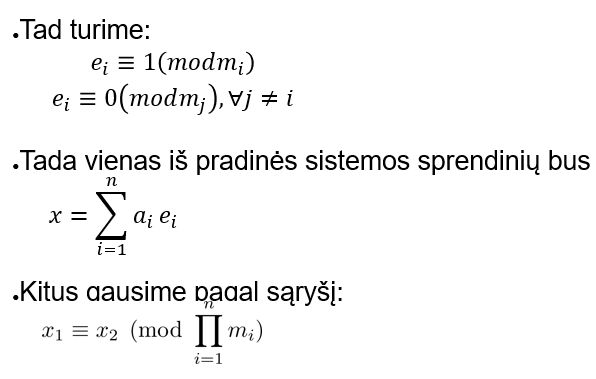
* 1. **Lyginių sistemos sprendimas.**
     1. **Lyginių sistemos sudarymas pagal uždavinį ir jos sprendimas.**











**Ir sita reiktu pasiaiskint 😊**

1. **Oilerio fi funkcija.**
   1. **Apibrėžimas, apskaičiavimas, savybės.**

**Oilerio fi funkcija** – tokia funkcija, kuri kiekvienam skaičiui grąžina skaičių už jį nedidesnių natūrinių skaičių, tarpusavyje pirminių su juo:



Beje,



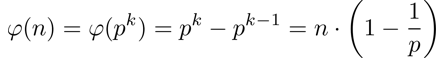
Dėl šio atvejo sakoma „nedidesnių“, o ne „mažesnių“.

**Apskaičiavimas**:

* Jei p yra pirminis ir



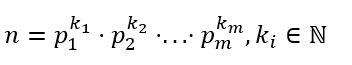
* Tai



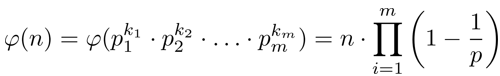
* Taip yra, nes iš n skaičių nuo 1 iki n iš p dalūs (ir su n ne tarpusavyje pirminiai) bus

p^k-1: p, 2p, 3p…

* Jei p1, p2, ..., pm yra pirminiai ir



* Tai



Oilerio teorema teigia, kad jei n ir a yra tarpusavyje pirminiai:



Oilerio funkcija naudojama RSA kriptosistemoje.

1. **Viešojo rakto kriptosistemos.**

Viešojo rakto kriptografijos atveju užšifravimui naudojami skirtingi raktai, vienas iš jų yra viešas, o kitas – slaptas.

Jei mes norime nuslėpti pranešimo turinį, slaptas bus iššifravimui naudojamas raktas, o jei norime autentifikuoti siuntėją, slaptas bus užšifravimui naudojamas raktas.

Viešojo rakto kriptografijai naudojami asimetrinio rakto algoritmai.

* 1. **Viešojo rakto kriptosistemų taikymai.**
* Yra sukurtos viešojo rakto kriptosistemos, kurios remiasi:
  + Skaičių skaidymu pirminiais dauginamaisiais
  + Diskrečiais logaritmais
* T.y. sprendimu lygčių, kurių forma



* + „eliptinėmis kreivėmis“
* Pačios kreivės nusakomos lygtimi:



* Kartu apibrėžiama operacija pagal tai, kaip tiesė eina per tokios kreivės taškus.

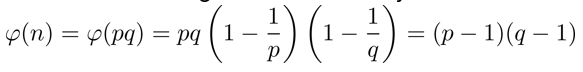
1. **RSA**

RSA yra viena iš viešojo rakto kriptosistemų.

* 1. **RSA rakto generavimas.**
* Pasirenkami du pirminiai skaičiai p ir q
* Jie sudauginami – taip gaunama viešojo rakto dalis



* Randama sandaugos Oilerio fi funkcijos reikšmė



* Pasirenkamas skaičius e, tarpusavyje pirminis su šia reikšme (antrojo viešojo rakto dalis)
* Apskaičiuojamas slaptas raktas – e moduliu φ(n) atvirkštinis skaičius (kuris, randamas išplėstiniu Euklido algoritmu)



* + 1. **Sugeneruoti RSA raktą.**
  1. **Užšifravimas ir dešifravimas RSA.**
* Pranešimą (skaičių) t, kurį norime nuslėpti, užšifruojame viešuoju raktu (n ir e) keldami jį e laipsniu ir imdami dalybos iš liekaną



* Iššifruojame slaptu raktu (analogiškai)



Teoriškai galime parinkti bet kokią e reikšmę, jei tik ji ir rastoji Oilerio funkcijos reikšmė bus tarpusavyje pirminiai skaičiau. Tačiau dažniausiai praktikoje naudojama e reikšmė yra 65537.

* + 1. **Užšifruoti ar dešifruoti pranešimą RSA.**
  1. **RSA kriptoanalizė.**

Šifro saugumas remiamas dviem hipotezėm:

* Išskaidyti skaičių pirminiais dauginamaisiais yra „sunku“ (eksponentinis asimptotinis laikinis sudėtingumas).
* Nulaužti RSA šifrą yra tiek pat sunku, kaip ir išskaidyti skaičių dauginamaisiais.

Nė viena iš šių hipotezių nėra įrodyta, bet kol kas niekas „paprasto“ (polinominio) algoritmo nesukūrė.

Išskaidymo dauginamaisiais taikymas RSA kriptoanalizei:

* Turime du viešąjį raktą sudarančius skaičius (n ir e)
* Iš jų n yra pirminių p ir q sandauga
* Jei kaip nors gautume n išskaidymą į p ir q, galėtume pakartoti visą rakto generavimą, taip gaudami ir slaptą raktą d

Kitokie kriptoanalizės būdai:

* Jei e mažas, gali būti, kad užšifruojamo skaičiaus laipsnis nebus didesnis už modulį, tada reikės tik ištraukti šaknį.
* Jei pats pranešimas perduotas užšifruotas kelis kartus su skirtingais moduliais, bet tuo pačiu e, galima jį atkurti panaudojus kinų teoremą.
  + 1. **Atlikti nesudėtingą kriptoanalizę (ne pagal kinų teoremą).**