



**Vilniaus
universitetas**

Informatikos ir informatinio mąstymo mokomoji veikla

Rikiavimo algoritmai

Mokyklos pedagogika



Kuriame
Lietuvos ateitį
2014–2020 metų
Europos Sąjungos
fondų investicijų
veiksmų programa



**Vilniaus
universitetas**

Informatikos ir informatinio mąstymo mokomosios veiklos sukurtos įgyvendinant projektą „Aukštųjų mokyklų tinklo optimizavimas ir studijų kokybės gerinimas Šiaulių universitetą prijungiant prie Vilniaus universiteto“ (Nr. 09.3.1-ESFA-V-738-03-0001), finansuojamą iš Europos socialinio fondo lėšų pagal 2014–2020 metų Europos Sąjungos fondų investicijų veiksmų programos 9 prioriteto „Visuomenės švietimas ir žmogiškųjų išteklių potencialo didinimas“ įgyvendinimo priemonę Nr. 09.3.1-ESFA-V-738 „Aukštųjų mokyklų tinklo tobulinimas“.

Metodinė medžiaga „Rikiavimo algoritmai“ skirta Mokyklos pedagogikos studijų programos moduliui „Informatikos didaktika“. Tikslinė grupė – būsimi informatikos pagrindinio ar vidurinio ugdymo mokytojai. Medžiaga siejasi su Bendrosiomis informatikos ir matematikos programomis, algoritmų ir programavimo pasiekimų sritimi. Atlikdami veikloje numatytas užduotis mokiniai išsiaiškins, kokie yra rikiavimo algoritmai, kaip jie veikia. Aptariami išrinkimo, greitasis, burbulo, įterpimo ir sąlajinis rikiavimo metodai. Pateikiamas kiekvieno rikiavimo algoritmo teorinis paaiškinimas ir įvairios užduotys.

Šioje informatikos ir informatinio mąstymo veikloje dalis medžiagos panaudota iš tarptautinio informatikos mokymo tinklo „Informatika be kompiuterio“ (*Computer Science Unplugged: csunplugged.org*), kurio visi ištekliai pateikiami taikant *Creative Common* licenciją.

Šios veiklos rengėjai: dr. Eglė Jasutė ir prof. dr. Valentina Dagienė

Visoms iliustracijoms yra taikoma Creative Commons Attribution-ShareAlike 4.0 International license (CC BY-NC-SA 4.0)

Redagavo: Viktoras Dagys

Išvadas

Rikiavimas, arba kartais dar vadinama rūšiavimu, yra viena pagrindinių kompiuterio atliekamų operacijų, sakoma, kad ketvirtadalis viso skaičiavimo laiko kompiuteris skiria rikiavimui. Rūšiavimas yra platesnis terminas, taikomas skirstyti objektus pagal kokią nors klasifikaciją, pavyzdžiui, sėklas pagal jų formą. Rikiavimu vadiname tai, ką galime sutvarkyti pagal eilę, remdamiesi skaičių seka arba abėcėle. Pavyzdžiui, rikiuojame objektus nuo trumpiausio iki ilgiausio, arba vardus pagal abėcėlę. Rikiavimas ir paieška neretai yra sudėtinė kitų algoritmų dalis.

Išrikiuotame sąraše daug lengviau surasti reikiamą informaciją. Telefonų knygoje, knygų kartotekoje elementai rikiuojami pagal abėcėlę. Jei jie nebūtų išrikiuoti pagal abėcėlę, būtų daug sunkiau susigaudyti informacijos gausybėje ir rasti reikiamą pavardę telefono adresatų sąraše ar autorių kartotekoje. Išrikiuotame skaičių sąraše lengva pamatyti didžiausią, mažiausią ar vienodus elementus.

Rikiavimas – tai algoritmas, sutvarkantis duomenis tam tikra sutarta tvarka. Algoritmo darbas priklauso nuo duomenų tvarkos apibrėžimo, duomenų struktūros, atminties panaudojimo rikiavimui, duomenų pateikimo viena laikiškumo, eiliškumo ir kitų veiksnių.

Sukurta daug rikiavimo metodų. Vieni lėtesni, kiti – greitesni, kai kurie priklauso nuo to, kokius duomenis turime, kokio galingumo kompiuterius naudojame. Kai kurie rikiavimo metodai yra neefektyvūs net ir atliekant juos kompiuteriu. Žinoma, yra atrasti keli greitesni rikiavimo metodai.

Štai keletas dažniausiai naudojamų rikiavimo algoritmų:

1. Išrinkimo algoritmas (angl. *Selection sort*)
2. Burbulo algoritmas (angl. *Bubble sort*)
3. Įterpimo algoritmas (angl. *Insertion sort*)
4. Rikiavimas Šelo metodu (angl. *Shell sort*)
5. Greitasis rikiavimas (angl. *Quick sort*)
6. Sąlajinis rikiavimas (angl. *Merge sort*)
7. Išorinis rikiavimas (angl. *External sorting*)
8. Piramidinis rikiavimas (angl. *Heap sort*)

Rikiuojant duomenis paprastai pasitelkiami kompiuteriai, todėl informatikai ieško vis greitesnio, patogesnio rikiavimo algoritmo. Dauguma rikiavimo algoritmų, pavyzdžiui,

išrinkimo, įterpimo ar burbulo, gerai tinka sprendžiant tik tam tikrus uždavinius. Greitojo ar sąlyginio rikiavimo algoritmai yra daug efektyvesni ir pritaikomi didesniam objektų skaičiui. Pavyzdžiui, 100 000 objektų greitojo rikiavimo algoritmas išrikiuoja 2000 kartų greičiau nei rikiuojant išrinkimo algoritmu, o 1 milijonų objektų – net 20 000 kartų greičiau.

Kompiuteris apdoroja milijonus duomenų, pavyzdžiui, daugybę tinklalapiuose besilankančių vartotojų savybių, iš vaizdo kamerų gautas nuotraukas – o jose, vėlgi milijonai pikselių. Rikiuojant šiuos duomenis greituoju ir sąlyginiu algoritmu, laiko skirtumai būtų akivaizdūs: pirmasis apdorotų duomenis per 1 sekundę, o antrajam tiems patiems duomenis apdoroti reikėtų 5 valandų. Prireiktų ne tik ilgesnio laiko, bet ir energijos sunaudojimas išaugtų 20000 kartų daugiau. Laiko ir energijos sąnaudos neturi įtakos programinei įrangai, tačiau yra svarbios naudojant prietaisus su baterijomis (planšetinius ar nešiojamuosius kompiuterius, mobiliuosius telefonus ir kt.).

Greitojo rikiavimo algoritmas gerai žinomu „skaldyk ir valdyk“ principu. Objektai padalijami į mažesnes grupes, kiekvienai iš jų atskirai taikomas greitojo rikiavimo algoritmas. Algoritmas kartojamas skaidarnt ir rikiuojant grupes, kol jose lieka po vieną objektą. Paprasta rikiuoti, kai lieka tik po vieną objektą!

Iš tikrųjų greitojo rikiavimo algoritmas yra daug greitesnis nei kiti rikiavimo metodai. Jis remiasi *rekursijos* idėja, pagal kurią algoritmas keletą kartų kreipiasi į save patį, kol išsprendžia uždavinį.

Internete galima rasti daug vaizdo įrašų apie rikiavimo metodus. Vienas populiariesnių – Davido Martino sukūrėta kai kurių populiarių rikiavimo algoritmų vizualizacija:

<http://www.sorting-algorithms.com/> (pateiktas titulinis šio tinklalapio vaizdas).



1 uždutis. Atverkite nurodytą tinklalapį, pasirinkite 2-3 rikiavimo metodus ir juos atidžiai peržiūrėkite. Kas labiausiai patiko? Ką geriausiai įsiminėte?

1 veikla. Lengviausias ir sunkiausias

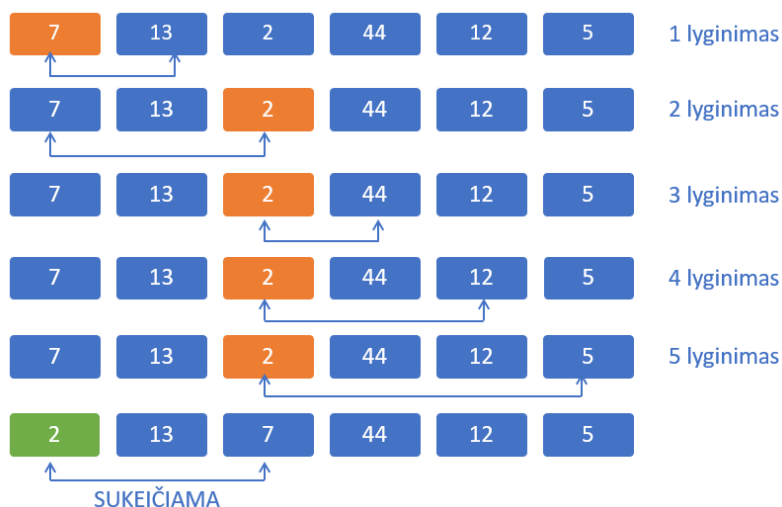
Kompiuteriu dažnai reikia išrikiuoti duomenis. Paprastai kompiuteris gali **palyginti dvi reikšmes vienu metu**.

Išrinkimo rikiavimo algoritmas

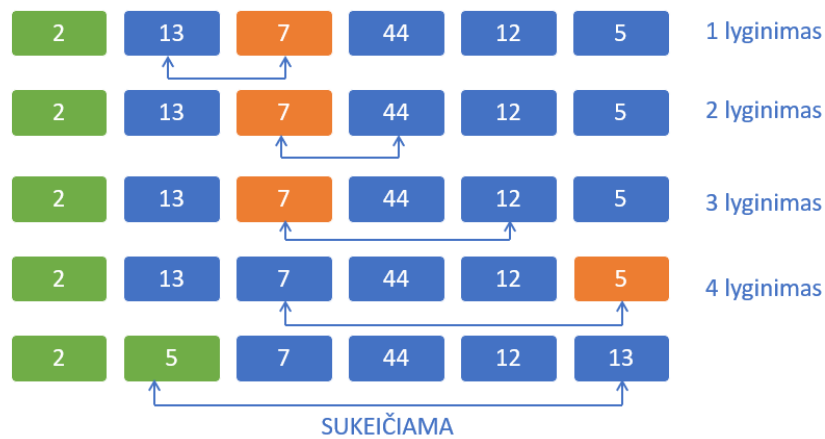
Vienas iš rikiavimo algoritmų yra **išrinkimo rikiavimo** (angl. *selection sort*) **algoritmas** (dar vadinamas išrenkamuoju rikiavimu). Išrinkimo algoritmas yra vienas paprastesnių, tačiau jis efektyvus tik tada, kai rikiuojamų objektų nėra daug arba kai objektai beveik surikiuoti, t. y. kai žinoma, jog tik keletas objektų yra ne savo vietoje. Kitais atvejais šis rikiavimas užims labai daug laiko.

Išrinkimo rikiavimo algoritmas grįstas dviem etapais: mažiausio elemento išrinkimas ir jo perkėlimas.

Schemoje pavaizduotas mažiausio elemento išrinkimas ir perkėlimas į sekos pradžią. Pirmajam elementui nustatyti buvo atlikti 5 lyginimai:



Toliau algoritmas kartojamas nuo antrojo sekos nario: išrenkamas mažiausias iš likusiųjų elementų ir perkeliamas į antrojo elemento vietą:



Algoritmas kartojamas, kol visi elementai išrikiuojami dėdjančia tvarka:



Kaip atliekamas išrinkimo algoritmas, kaip naudojamas objektų svėrimas svarstyklėmis, parodyta vaizdo įrašė („CS unplugged“):

https://www.youtube.com/watch?v=cVMKXKoGu_Y&ab_channel=UCComputerScienceEducation

Diskusija

Pažiūrėkite įrašą ir aptarkite, ar pakankamai aiškiai parodytas objektų išrinkimas ir keitimas. Ar duotumėte šią užduotį mokiniams? Kurios klasės mokiniams būtų tinkamas šis pavyzdys?

Toliau pateikiame dvi veiklas, kurios skiriamos mokiniams (du darbo lapai). Vienai veiklai reikia fizinių priemonių (daiktų, svirtinių svarstyklių), kitai – užtenka popierinio darbo lapo.

Užduotis. Įsigilinkite į abi veiklas. Kurią Jūs rinktumėtės savo mokiniams? Kurio amžiaus mokiniams būtų tinkama šios veiklos?

A veikla

Priemonės

- 8 vienodo dydžio, bet skirtingo svorio pakuotės (gali būti pakuotės nuo pieno, nepermatomi plastikiniai buteliukai nuo gėrimo ar skardinės dėžutės, pripildytos smėlio ar vandens).
 - Svirtinės svarstyklės.
1. Mokiniai suskirstomi į grupes.
 2. Kiekvienai grupei duodamas darbo lapas „Svorių rikiavimas“, sveriamų pakuočių rinkiniai ir svirtinės svarstyklės.
 3. Mokytojas paaiškina darbo atlikimo eigą.
 4. Mokiniai atlieka šią veiklą ir tada aptaria rezultatus.

B veikla

1. Mokiniai suskirstomi poromis.
2. Kiekvienai porai duodamas darbo lapas „Bebras rikiuoja“.
3. Mokiniai atlieka užduotis, palygina rezultatus, aptaria algoritmo žingsnius.

Pastaba: „Bebro“ užduočių darbo lapuose mokiniai gali rikiuoti ir fizinius objektus: vietoj rąstų naudoti skirtingo ilgio pieštukus, domino kauliukus.

1 darbo lapas: Sviurių rikiavimas

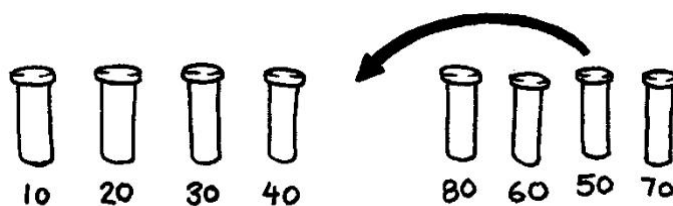
Ką daryti:

1. Į kiekvieną pakuotę įpilkite skirtingą kiekį smėlio arba vandens. Sandariai uždarykite.
2. Sumaišykite pakuotes, kad nežinotumėte, kiek jose smėlio ar vandens.
3. Suraskite lengviausią pakuotę. Kokiu paprasčiausiu būdu tai galima padaryti?

Patarimai. Naudokite svirtines svarstyklės. Vienu metu jomis galite palyginti tik du svorius.

1. Atsitiktinai pasirinkite 3 pripildytas pakuotes. Išrikiuokite jas nuo lengviausios iki sunkiausios pasverdami svirtinėmis svarstyklėmis. Kaip tai atlikote? Kiek mažiausiai kartų reikėjo sverti? Kodėl?
2. Išrikiuokite visas 8 pripildytas pakuotes nuo lengviausios iki sunkiausios.
3. Tada patikrinkite, ar gerai išrikiavote, lygindami dviejų gretimų pakuočių svorį svirtinėmis svarstyklėmis.

Pirma, surandamas mažiausias svoris ir atidedamas į šoną. Antra, surandamas mažiausias svoris iš likusiųjų ir atidedamas į šoną. Tai kartojama, kol visi svoriai išrikiuojami.



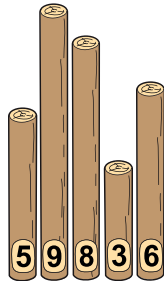
Suskaičiuokite, kiek kartų lyginote svorius.

Mąstykite. Parodykite, kaip matematiškai galima suskaičiuoti, kiek kartų reikia lyginti 8 objektus norint juos išrikiuoti pagal svorį. Kiek kartų reikia lyginti 9 objektus, kai norima juos išrikiuoti? 20 objektų?

2 darbo lapas: Rąstų rikiavimas

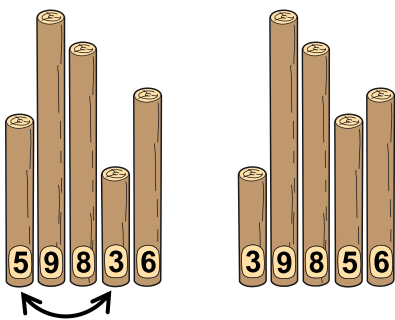
Išrinkimo algoritmo taikymo pavyzdys rąstams rikiuoti.

Norima surikiuoti rąstus nuo trumpiausiojo iki ilgiausiojo.

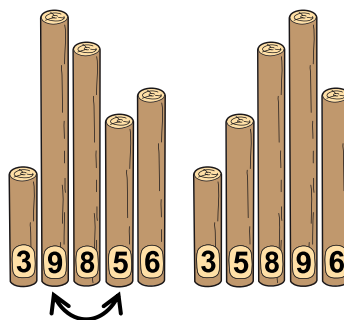


Rikiavimas vyksta lyginant du gretimus rąstus. Taip bebras lygindamas su paskesniu išrenka trumpiausią rąstą.

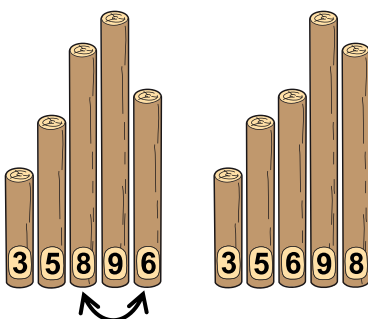
Trumpiausias rąstas sukeičiamas vietomis su pradžioje esančiu (5 sukeičiamas su 3).



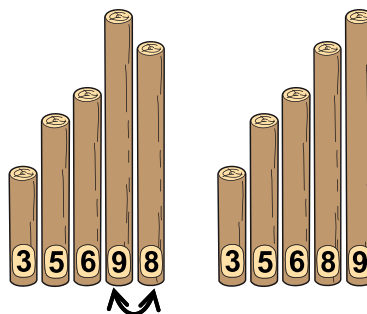
Tada antrasis rąstas sukeičiamas su toliau esančiu trumpiausiu rąstu (5 sukeičiamas su 9).



Toliau lyginami likusieji rąstai ieškant trumpiausio rąsto tuo pačiu būdu. Radę sukeičiame jį su trečioje vietoje esančiu rąstu (6 sukeičiamas su 8).



Tada trumpiausias rąstas iš likusių sukeičiamas su ketvirtoje vietoje esančiu rąstu (8 sukeičiamas su 9).

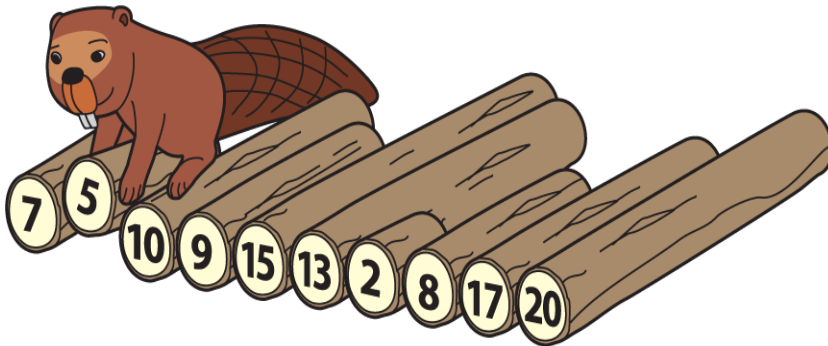


Papildomos užduotys

Išspręskite uždavinius naudodami išrinkimo rikiavimo algoritmą.

Rąstų rikiavimas

Bebras nori išrikiuoti rąstus pagal ant jų nurodytus skaičius.

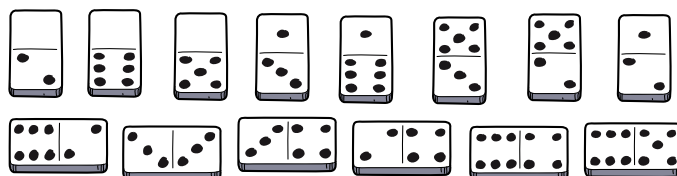


Kiek rąstų bebras turės perkelti, kad išrikiuotų šią krūvą nuo mažiausio iki didžiausio?

„Perkelti“ reiškia VIENĄ rąstą pakelti iš vienos vietos ir jį padėti į kitą vietą.

Domino kauliukų rikiavimas

Gertrūda nori išdėlioti domino kauliukus pagal kiekvieno jų taškų skaičių nuo mažiausios reikšmės iki didžiausios.



Kiek domino kauliukų ji turės perkelti, kol išdėlios visus didėjančiai?

2 veikla. Lengviausias ir sunkiausias

Greitojo rikiavimo algoritmas

Greitojo rikiavimo (angl. *quicksort*) algoritmas yra daug greitesnis ir paprastesnis už išrinkimo algoritmą, todėl labai dažnai naudojamas.

Algoritmo idėja. Tarkime turime sąrašą, imame jo vidurinį jo elementą (7):

[1, 8, 54, 2, 3, 7, 34, 99, 2, 5, 12].

Visus elementus, mažesnius už 7 perrašome kairėje pusėje, visus didesnius – dešinėje:

[1, 2, 3, 2, 5, 7, 8, 54, 34, 99, 12].

Sudarome du posraščius ir tą patį metodą taikome tiems dviem posraščiams

[1, 2, 3, 2, 5] ir [8, 54, 34, 99, 12]:

[1, 2, 2, 3, 5, 7, 8, 12, 34, 54, 99].

Vėl taikome naujiems posraščiams: [1, 2, 2], [8, 12], [54, 99]:

[1, 2, 2, 3, 5, 7, 8, 12, 34, 54, 99].

Iki kol sąrašas tampa išrikiuotas (posraščiai tik iš 1 elemento):

[1, 2, 2, 3, 5, 7, 8, 12, 34, 54, 99].

Vėl pateiksime veiklą naudodami svėrimą.

Greitasis rikiavimas: svėrimų veikla

Priemonės

8 vienodo dydžio, bet skirtingo svorio pakuotės (gali būti pakuotės nuo pieno, nepermatomi plastikiniai buteliukai nuo gėrimo ar skardinės dėžutės, pripildytos smėlio ar vandens)

Svirtinės svarstyklės

1. Mokiniai suskirstomi į grupes.
2. Kiekvienai grupei duodamas darbo lapas „Skaldyk ir valdyk“, sveriamų pakuočių rinkiniai ir svirtinės svarstyklės.
3. Mokytojas paaiškina darbo atlikimo eigą.
4. Mokiniai atlieka šią veiklą ir tada aptaria rezultatus.

Peržiūrėkite greitojo rikiavimo atlikimo vaizdo įrašą („CS unplugged“, nuo 2:10):

https://www.youtube.com/watch?v=cVMKXKoGu_Y&ab_channel=UCComputerScienceEducation

Diskusija

Pažiūrėkite įrašą ir aptarkite, ar pakankamai aiškiai parodytas naujų posąrašių sudarymas, elementų rikiavimas. Ar duotumėte šią užduotį mokiniams? Kurios klasės mokiniams būtų tinkamas šis pavyzdys?

3 darbo lapas: Skaldyk ir valdyk

Ką daryti:

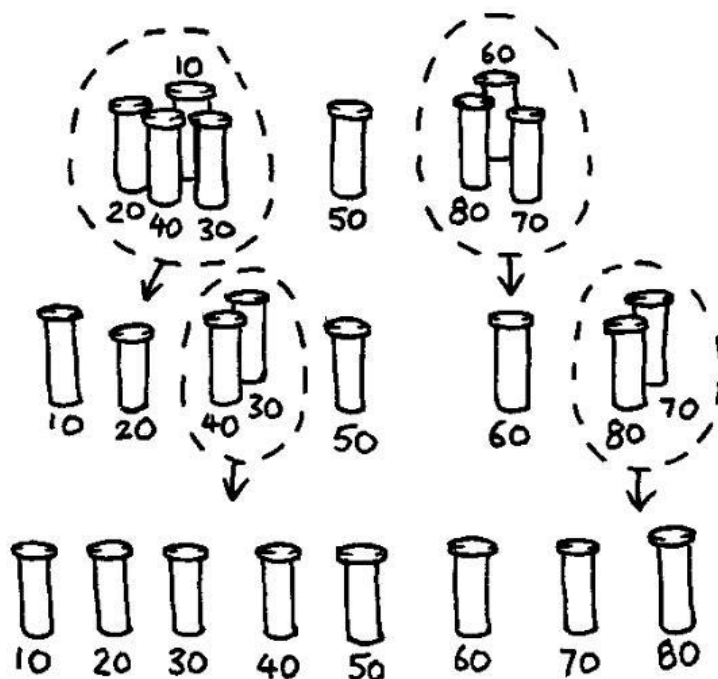
1. Į kiekvieną pakuotę įpilkite skirtingą kiekį smėlio arba vandens. Sandariai uždarykite.
2. Sumaišykite pakuotes, kad nežinotumėte, kiek jose smėlio ar vandens.

Atsitiktinai pasirinkite vieną iš pakuočių ir padėkite ją ant svarstyklių lėkštelės.

Visas likusias pakuotes palyginkite su atsitiktinai pasirinkta pakuote. Lengvesnes už ją padėkite kairėje svarstyklių pusėje, o sunkesnes – dešinėje. Pakuotę, su kuria lyginate kitas pakuotes, padėkite viduryje tarp grupių. (Baigus sverti visas pakuotes, gali būti taip, kad vienoje pusėje bus daug daugiau pakuočių nei kitoje.)

Pasirinkite vieną iš atrinktų pakuočių grupių ir pakartokite tą patį: atsitiktinai iš grupės pasirinkite pakuotę ir palyginkite visas likusias pakuotes su ja. Tą patį padarykite ir su kita pakuočių grupe. Pakuotę, su kuria lyginate kitų pakuočių svorius, padėkite viduryje tarp sudarytų naujų dviejų grupių.

Kartokite šiuos veiksmus, kol grupėse liks tik po vieną pakuotę. Kadangi visose pakuočių grupėse liko tik po vieną objektą, vadinasi, visi objektai yra išrikiuoti nuo lengviausio iki sunkiausio.



Pagalvokite, aptarkite. Kiek kartų reikėjo lyginti pakuotes naudojant šį algoritmą?

Greitasis rikiavimas yra daug efektyvesnis už *išrinkimo algoritmą*, jei iš pradžių nepavyksta pasirinkti lengviausio ar sunkiausio objekto. Jei pasirenkamas vidutinio svorio objektas, taikant greitąjį rikiavimą užtenka 14 kartų palyginti visus objektus ir taip juos išrikiuoti nuo lengviausio iki sunkiausio. Padaryti tą patį taikant *išrinkimo algoritmą* prireiktų 28 lyginimo kartų.

Pamąstykite Kiek kartų greitojo rikiavimo algoritmu reikėtų lyginti objektus, jei visada atsitiktinai būtų pasirenkamas lengviausias objektas?

Burbulo rikiavimo algoritmas

Burbulo rikiavimo (angl. *bubble sort*) **algoritmo** principas – nuosekliai iš eilės patikrinti gretimų objektų poras, prireikus objektus sukeisti, t. y. perkelti mažesnį arčiau pradžios. Šitaip per pirmą žingsnį lengviausias objektas perkeliamas į pirmą poziciją, vėliau tas pats principas taikomas likusiems objektams ir t. t.

Panagrinėkite paprastą pavyzdį. Rikiuokime penkis skaičius: 5 1 4 2 8 nuo mažiausio iki didžiausio.

Pirma lyginimų peržiūra:

(5 1 4 2 8) (1 5 4 2 8) Čia algoritmas palygina pirmus 2 elementus ir apkeičia juos vietomis.

(1 5 4 2 8) (1 4 5 2 8)

(1 4 5 2 8) (1 4 2 5 8)

(1 4 2 5 8) (1 4 2 5 8) Elementai išdėstyti tinkama tvarka, todėl keisti nereikia.

Antra lyginimų peržiūra:

(1 4 2 5 8) (1 4 2 5 8)

(1 4 2 5 8) (1 2 4 5 8)

(1 2 4 5 8) (1 2 4 5 8)

(1 2 4 5 8) (1 2 4 5 8)

Dabar elementai išdėstyti tinkama tvarka, bet algoritmas dar to nežino. Algoritmui reikia vienos peržiūros be pakeitimų, tada žinos, kad elementai išdėstyti reikiama tvarka.

Trečia lyginimų peržiūra:

(1 2 4 5 8) (1 2 4 5 8)

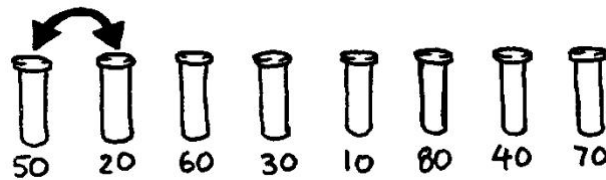
(1 2 4 5 8) (1 2 4 5 8)

(1 2 4 5 8) (1 2 4 5 8)

(1 2 4 5 8) (1 2 4 5 8)

Galiausiai algoritmas baigia darbą.

Burbulo algoritmo veikimo principas primena virimą, kai oro burbulai kyla į paviršių, dėl to jis ir vadinamas burbulo rikiavimo metodu. Burbulo rikiavimo metodas yra vienas iš paprastų, bet nelabai efektyvių rikiavimo algoritmų.



Pažiūrėkite burbulo algoritmo vieną ar kitą vaizdo įrašą:

https://www.youtube.com/watch?v=dKExQ7_ylko&ab_channel=%E5%A4%8F%E7%B6%B2%E4%B8%8A%E6%95%99%E5%AE%A4

https://www.youtube.com/watch?v=k6QYyQfdwdg&ab_channel=MikeGame

Toliau pateikiame dvi veiklas mokiniams. Pirmoji veikla fizinė, antrajai užtenka popierinio darbo lapo.

Apmąstykite. Įsigilinkite į abi veiklas. Kurią Jūs rinktumėtės savo mokiniams? Kurio amžiaus mokiniams būtų tinkama kiekviena iš šių veiklų?

A veikla

1. Mokiniai suskirstomi į dvi grupes (po 10–13).
2. Mokytojas paaiškina burbulo rikiavimo algoritmą, parodo pavyzdžių.
3. Vienos grupės mokiniai išsimaišo pagal ūgį ir sustoja į vieną eilę.
4. Kitai grupei duodamas darbo lapas „Išsirikiuokite pagal ūgį“. Kiekvieną rikiavimo etapą atlieka vis kitas grupės mokinys.
5. Mokiniai atlieka nurodytą algoritmą ir aptaria rezultatą.
6. Grupės gali apsikeisti ir kita grupė gali išbandyti šį rikiavimo algoritmą.

B veikla

1. Mokiniai suskirstomi poromis.
2. Kiekvienai porai duodamas darbo lapas „Rikiavimo uždaviniai“.
3. Mokiniai atlieka užduotis, palygina rezultatus, aptaria algoritmo žingsnius.

Galite mokiniams duoti sugalvoti savo uždavinių ir pateikti draugams spręsti.

3 darbo lapas: Išsirikiuokite pagal ūgį

Išrikiuoti mokinius pagal jų ūgį



1. Rikiuosite nuo žemiausio (kairėje) iki aukščiausio (dešinėje) taikydami burbulo algoritmą:
a. Palyginkite pirmus du mokinius:



- b. Jei pirmas yra aukštesnis, tai jis susikeičia vietomis su antruoju:



- c. Toliau vėl lyginama su gretimu ir susikeičia su juo, jei yra aukštesnis:



- d. Lyginama ir keičiama tol, kol mokinys atsiduria savo vietoje. Pirmas rikiavimo etapas baigtas:



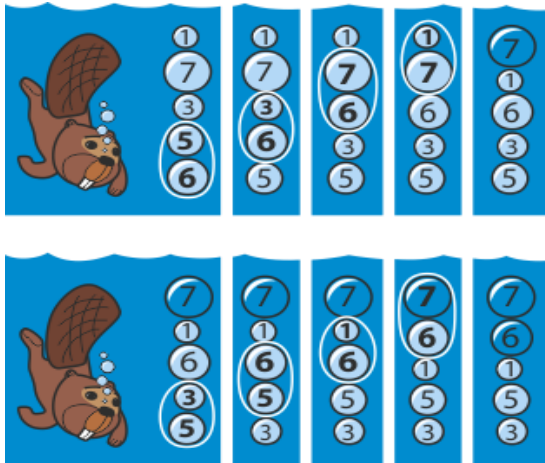
- e. Rikiavimo žingsniai kartojami nuo pradžių, kol visi mokiniai išsirikiuojami pagal ūgį.

Apmąstykite. Ar supratote kaip vyksta rikiavimas burbulo metodu? Paskaitykite apie burbulo metodą Vikipedijoje, yra pateikta pavyzdžių.

3 darbo lapas: Rikiavimo uždaviniai

Vandens burbulai

Bebras žaidžia su vandens burbulais: rikiuoja juos nuo mažiausio (apačioje) iki didžiausio taikydamas burbulo rikiavimo algoritmą.



Pradedama nuo apačios. Ima du burbulus ir lygina: jei didesnis yra apačioje, **sukeičia vietomis** ir vėl lygina su gretimu burbulu.

Lyginama ir keičiama tol, kol burbulas atsideduria **savo vietoje**.

Ir vėl **kartojama** iš pradžių (apačios).

Baikite rikiuoti bebro burbulus. Pavaizduokite, ką toliau turi daryti bebras.

Triušiai ir vėžliai



Zuikiai žaidžia su vėžliais. Gyvūnai nori kuo greičiau išsirikiuoti nuo mažiausio numerio iki didžiausio pagal burbulo rikiavimo algoritmą. Todėl pradeda zuikiai.

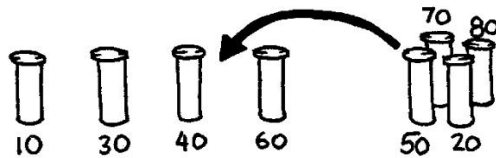
	Zuikis, stovintis B vietoje, pasilygino su A zuikiu ir susikeitė su juo vietomis.
	Paskui jis pasilygino su C zuikiu ir susikeitė vietomis.
	Zuikiai atliko du pasikeitimus, kol zuikis su numeriu 5 atsistojo į savo vietą.

Užduotys. Kiek pasikeitimų turi padaryti vėžliai, kad vėžlys su numeriu **8** atsistotų į savo vietą?

Kiek dar pasikeitimų turi atlikti gyvūnai, kad visi išsirikiuotų nuo mažiausio iki didžiausio numerio?

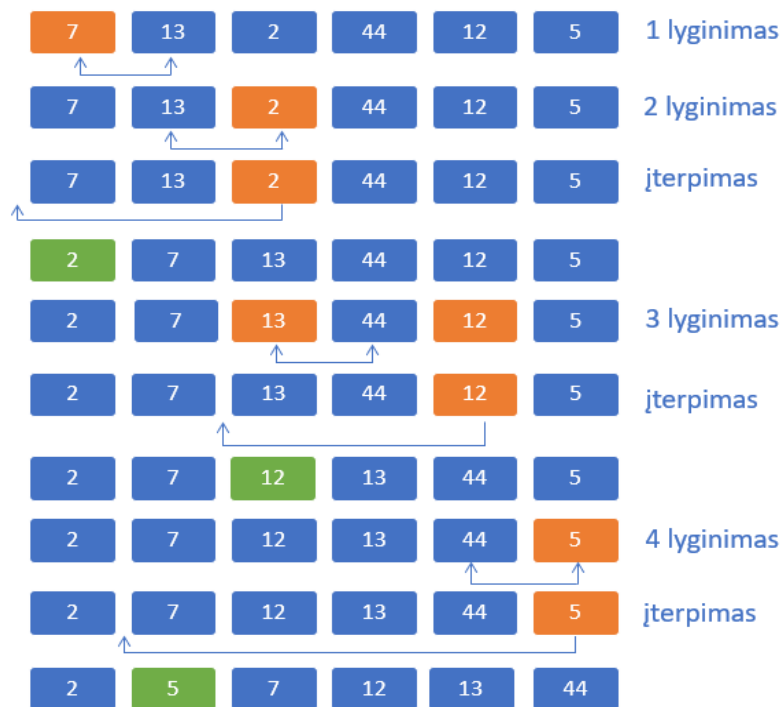
Įterpimo rikiavimo algoritmas

Pagrindinis įterpimo rikiavimo (angl. *insertion sort*) algoritmo principas – kiekvienas elementas imamas iš eilės ir įterpiamas į jam tinkamą vietą jau surikiuotoje objektų grupėje. Kiekvienąkart įterpus objektą, neišrikiuotoje grupėje objektų mažėja, o išrikiuotoje – daugėja.



Kai žmonės rankomis rikiuoja kortas, dauguma jų naudoja panašų metodą į įterpimo rikiavimo algoritmą.

Schemoje vaizduojamas įterpimo rikiavimo algoritmas. Pirmiausia lyginamas antrasis elementas su pirmuoju. Jei antrasis yra mažesnis, tai jis įterpiamas prieš pirmąjį. Toliau tikrinimas antroje pozicijoje esantis elementas su paskesniu, kol randamas mažesnis.



Peržiūrėkite įterpimo algoritmo vaizdo įrašą:

https://www.youtube.com/watch?v=bc5uT-tBvM&ab_channel=MikeGame

Diskusija. Ką galėtume pasakyti apie šį įrašą? Ar aišku, kaip rikiuojama? Paieškokite, gal rasite geresnį įterpimo algoritmo vaizdo įrašą.

Užduotis. Reikia išrikiuoti kortas (darbo lapas „Katinų kortų rikiavimas“) taikant įterpimo algoritmą.

5 darbo lapas: Kortų rikiavimas įterpimo metodu

Reikia išrikiuoti duotas kortas nuo mažiausios vertės iki didžiausios.

Naudokite įterpimo algoritmą:

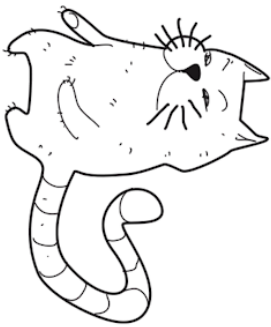
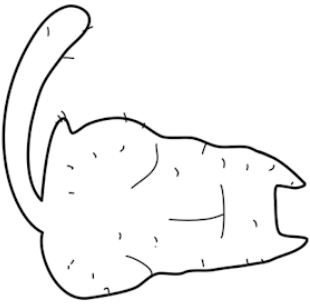
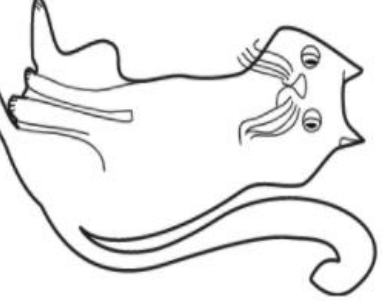
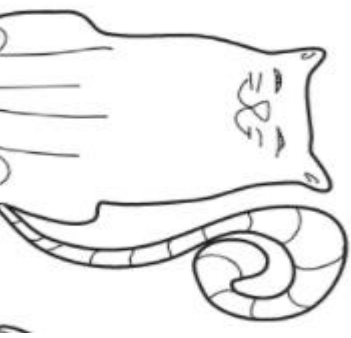
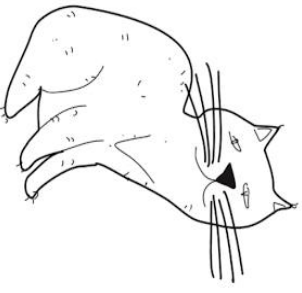
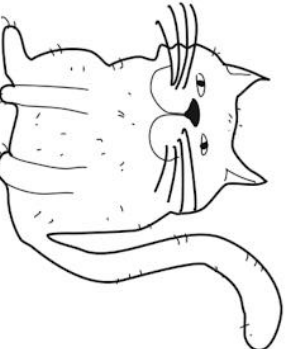
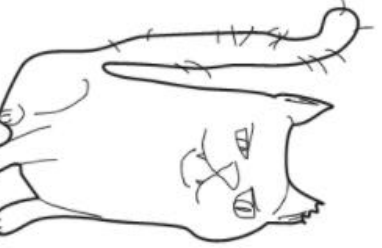
Paimkite pirmąją kortą iš neišrikiuotų kortų eilės ir perkeltkite (įterpkite) į vietą, kurioje ji turėtų būti pagal vertę.



Kartokite šį žingsnį, kol visos kortos bus surikiuotos.

Pagalvokite. Kiek kartų perkėlėte kortas?

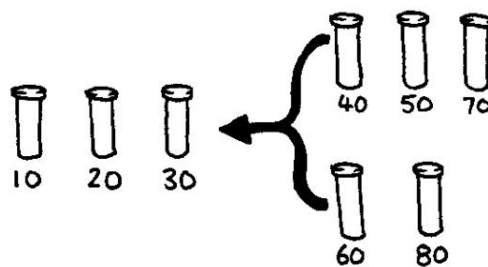
Apmąstykite. Išmaišytą 24 lošimo kortų kaladę išrikiuokite pagal kortų rūšis ir vertę naudodami įterpimo algoritmą. Kiek kortų perkėlimų atlikote?

54	
32	
2	
17	
40	
12	
10	
7	
25	
30	

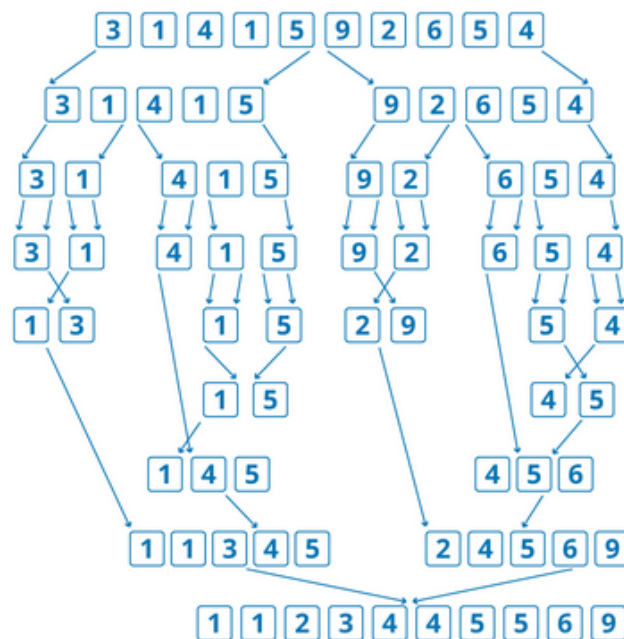
Sąlyginio rikiavimo algoritmas

Išskaidyk ir sujunk

Sąlyginio rikiavimo (angl. *merge sort*) algoritmas – vienas iš „skaldyk ir valdyk“ principu grindžiamų rikiavimo algoritmų. Pirmiausia objektai atsitiktinai padalijami į dvi lygias (arba panašias, jei yra nelyginis objektų skaičius) grupes. Kiekviena grupė rikiuojama atskirai, paskui abi grupės sujungiamos: lyginant skirtingų grupių objektus po du iš eilės mažiausias objektas perkeliamas į jungtinės objektų grupės pabaigą. Paveiksle 40 ir 60 gramų objektai yra kiekvienos grupės pradžioje, į jungtinę grupę dedamas mažesnio svorio objektas – 40 g ir t. t.



Sąlyginio rikiavimo algoritmo pavyzdys gali būti pavaiduotas štai tokia schema:



Išnagrinėkite šį pavyzdį ir pagal jį surikiuokite štai šiuos duomenis:

56 29 35 42 15 41 75 21

Peržiūrėkite sąlyginio algoritmo vaizdo įrašą:

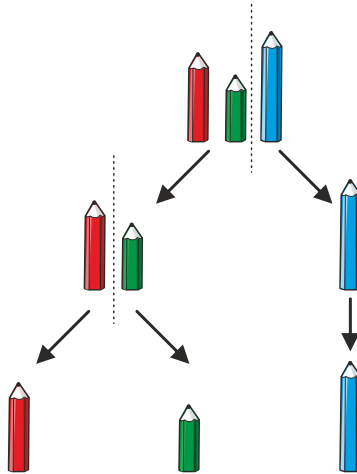
<https://www.youtube.com/watch?v=nSQL9YYvq8>

Diskusija. Ką galėtume pasakyti apie šį įrašą? Ar aišku, kaip rikiuojama? Paieškokite, gal rasite geresnį sąlyginio rikiavimo algoritmo vaizdo įrašą.

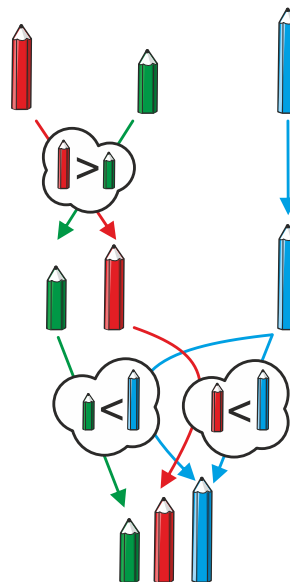
6 darbo lapas: Pieštukų rikiavimas sąlajiniu metodu

Iš savo penalo išsirinkite patį trumpiausią pieštuką.

1. Mokytojo padedami pasidalinkite į dvi apylyges grupes.
2. Kiekviena grupė dalijasi dar į dvi apylyges grupes.
3. Darbas kiekvienoje grupėje – išrikiuoti savo pieštukus nuo trumpiausio iki ilgiausio naudojant sąlajinį rikiavimą:
 - a. Dalinkite grupės pieštukus į apylyges dvi dalis, kol jie bus po vieną.



- b. Tuomet pieštukus vėl grupuojate lygindami po du ir dedate trumpesnį į kairę.



Galiausiai savo grupėje turite išrikiuotus pieštukus nuo trumpiausio iki ilgiausio.

Sujunkite visus klasės pieštukus ir išrikiuokite juos nuo trumpiausio iki ilgiausio.

Užomina: turite 4 išrikiuotų pieštukų grupes.

Sprendimas ir užuominos

Lengviausias ir sunkiausias

1. Geriausias būdas surasti lengviausią objektą yra tikrinti visus objektus lyginant vieną su kitu. Kitaip tariant, objektai lyginami po du, iš kiekvienos poros ant svarstyklių lėkštelės paliekamas lengvesnis objektas tol, kol perrenkami ir palyginami visi objektai.
2. Objektų svoris lyginamas svirtinėmis svarstyklėmis. Tai galima atlikti palyginus svorį triskart, o kartais ir dukart, jei mokiniai supranta, kad lyginimo operacija yra tranzityvi (t. y., jei A yra lengvesnis už B, o B yra lengvesnis už C, tai A turi būti lengvesnis už C).

Pamąstykite

Norint nustatyti mažiausio svorio objektą iš dviejų reikia vieną kartą juodu palyginti. Ieškant lengviausio iš trijų objektų reikia tuos objektus palyginti du kartus, iš keturių objektų – reikia juos palyginti tris kartus ir t. t. Rikiuojant aštuonis objektus ir norint iš jų surasti pirmą lengviausią reikia 7 kartus palyginti visus objektus, šešis kartus – norint surasti antrą lengviausią objektą ir t. t.

Išeina, kad reikia palyginti

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ kartus.}$$

Rikiuojant n objektų reikėtų palyginti $1 + 2 + 3 + 4 + \dots + n - 1$ kartų.

Šiuos skaičius sudėti yra paprasta, kai jie sugrupuojami.

Pavyzdžiui, sudedant skaičius $1 + 2 + 3 + \dots + 20$, sugrupuojama:

$$(1+20)+(2+19)+(3+18)+(4+17)+(5+16)+$$

$$(6+15)+(7+14)+(8+13)+(9+12)+(10+11)=21 \times 10 = 210$$

Bendru atveju gaunama $1 + 2 + 3 + 4 + \dots + n - 1 = n(n - 1)/2$.

Šaltiniai

Petrauskas, L., Skūpienė, J. Informatikos olimpiados: algoritmai ir taikymo pavyzdžiai. Vilnius 2006: <https://siom.lmio.lt/m/t/knyga.pdf>

Algorimai ir duomenų struktūros. 4 paskaita: http://web.vu.lt/mii/m.sabaliauskas/wp-content/uploads/2020/02/4paskaita_2020.pdf

Dagienė V., Grigas G., Augutis K. Šimtas programavimo uždavinių. Kaunas: Šviesa, 1986, 224 p.

Greitasis rikiavimo algoritmas. Vikipedija: <https://en.wikipedia.org/wiki/Quicksort>

Išrinkimo rikiavimo algoritmas. Vikipedija:

https://lt.wikipedia.org/wiki/I%C5%A1rinkimo_rikiavimo_algoritmas

Sqlajos rikiavimo algoritmas. Vikipedija: https://en.wikipedia.org/wiki/Merge_sort

5 Sorting Algorithms Explained to Kids: <https://codinghero.ai/5-sorting-algorithms-explained-to-kids/>

