Vilniaus universitetas

Informatikos ir informatinio mąstymo mokomoji veikla

Trumpiausio kelio paieška

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 „Trumpiausio kelio paieška“ skirta informatikos ir informacinio mąstymo mokymui. Tikslinė grupė – būsimi informatikos pagrindinio ugdymo mokytojai. Medžiaga siejasi su informatikos ir matematikos Bendrosiomis programomis, algoritmų ir programavimo pasiekimų sritimi. Atlikdami veikloje numatytas užduotis studentai išsiaiškins, kokiuose uždaviniuose ir kaip taikomas trumpiausio kelio algoritmas, susipažins su Dijkstros algoritmu trumpiausiam keliui grafe rasti. Pateikiamas teorinis temos pagrindimas mokytojui, aptariamos pagrindinės srities sąvokos.

Autoriai: dr. Eglė Jasutė ir prof. dr. Valentina Dagienė

Redagavo: Viktoras Dagys

Vilnius, 2022

Įvadas

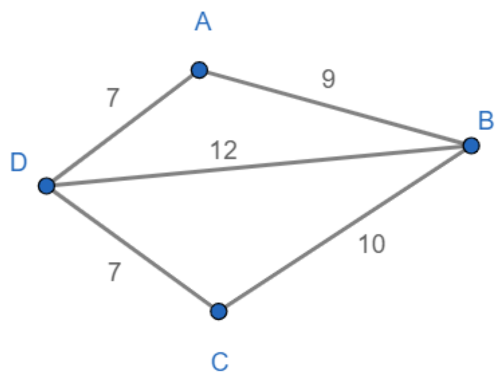
Dažnai tenka nuvykti iš vieno taško į kitą: iš namų į mokyklą, iš vieno miesto į kitą, iš vienos šalies į kitą. Vykti galima įvairiomis transporto priemonėmis: lėktuvais, traukiniais, autobusais, automobiliais, dviračiais, pėsčiomis. Mes dažnai ieškome, kaip tai padaryti greičiausiai ar pigiausiai.

Tokias gyvenimiškas ir kasdienes problemas šiandien padeda spręsti įvairios skaitmeninės priemonės („Google Maps“, trafi.lt, stops.lt, skrydžių, traukinių paieškos įrankiai ir pan.). Kaip jos veikia? Kokius algoritmus naudoja? Ši veikla skirta išnagrinėti vieną dažniausiai naudojamų planuojant maršrutus algoritmą – trumpiausio kelio paieškos grafe algoritmą.

Maršruto uždaviniai sprendžiami pasitelkus grafų teoriją. Grafas yra figūra, sudaryta iš taškų (vadinamų viršūnėmis) ir atkarpų (vadinamų briaunomis). Briauna vaizduojama bet kokia linija, nebūtinai tiesia (gali būti lenkta, banguota ir pan.), tačiau ji visada jungia dvi viršūnes. Kai briauna jungia viršūnę su ja pačia, ji vadinama kilpa. Grafuose linijų ilgis neturi reikšmės – nėra jokio mastelio. Grafus nagrinėjo ir jų teorijai pagrindą padėjo XVIII a. matematikas L. Oileris (Leanard Euler).

Dažnai tikslinga grafo briaunai (arba lankui) priskirti kokį nors dydį. Pavyzdžiui, jei grafu modeliuojame vietovės žemėlapį (viršūnėmis – miestus, o briaunomis – kelius), tai briaunoms galima priskirti tų kelių ilgius.

Grafas, kurio visoms briaunoms (lankams) yra priskirti dydžiai (svoriai), vadinamas svoriniu. Dažniausiai nagrinėjami svoriniai grafai, kurių briaunų svoriai yra skaičiai. Patyrinėkime svorinio grafo pavyzdį.



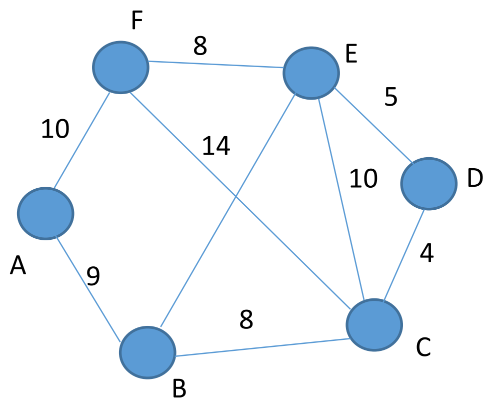
Yra keturios viršūnės: A, B, C ir D ir penkios briaunos. Briaunos turi svorius: AB svoris 9, AD – 7, BC – 10, CD – 7 ir BD – 12. Šis grafas neorientuotas, tai yra visai nesvarbu, ar nurodysime briauną, pavyzdžiui, AB, ar BA. Jei yra nurodytos briaunų kryptys, tada grafas vadinamas orientuotu ir briaunos turi būti nurodomos jų krypties tvarka.

Toliau susipažinsime su vienu iš populiariausių algoritmų trumpiausiam keliui rasti – vadinamuoju Dijkstros algoritmu.

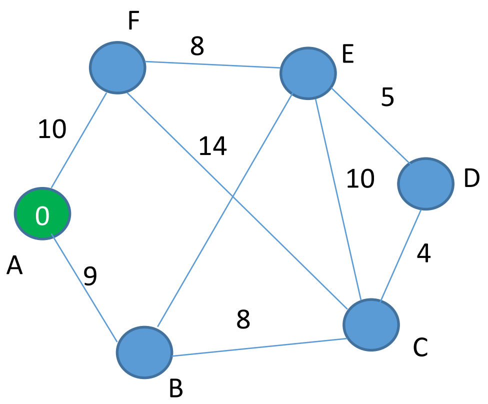
Dijkstros algoritmas trumpiausiam keliui rasti

Išnagrinėkime Dijkstros algoritmą pavaizduotam grafui.

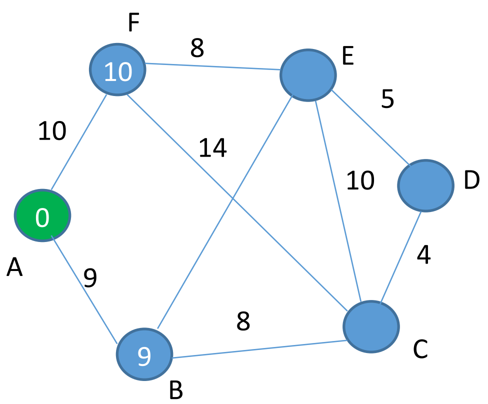
Surasime trumpiausius atstumus nuo viršūnės A iki kiekvienos kitos grafo viršūnės.



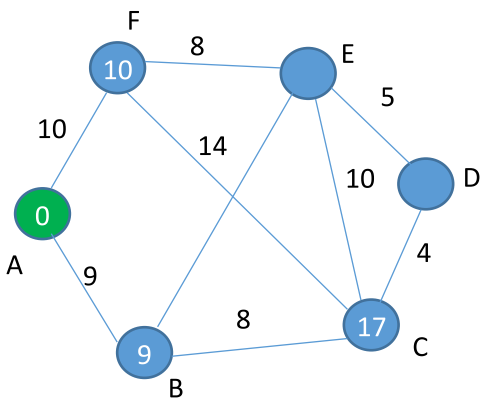
1. Pasirinkime pradinę viršūnę, nuo kurios ieškosime trumpiausių kelių iki kitų viršūnių. Tarkime, A. Šiai viršūnei priskirkime nulinį atstumą (0).



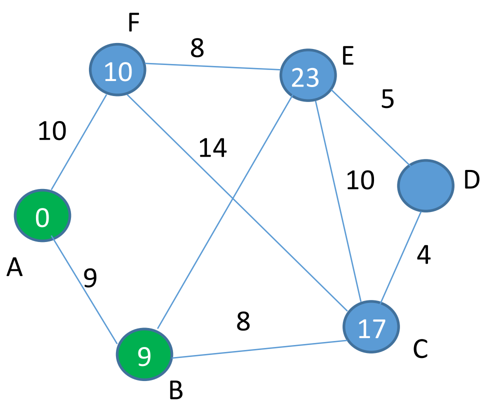
2. Viršūnės A kaimynystėje yra viršūnės B ir F (vadinamos gretimomis viršūnėmis). Pažymime atstumus iki jų.



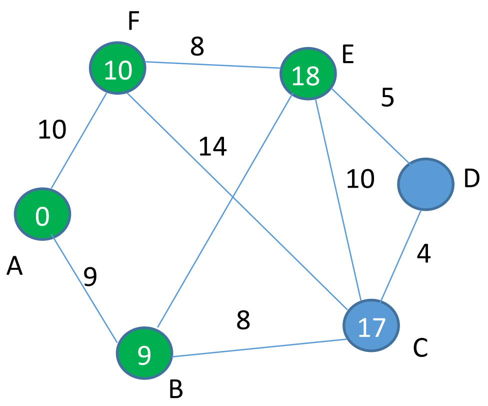
3. Pasirenkame viršūnę B ir pažymime atstumus iki jos gretimų viršūnių E ir C. Apskaičiuojame atstumą iki viršūnės C: prie atstumo BC pridedame atstumą AB ir užrašome atstumą viršūnėje C.



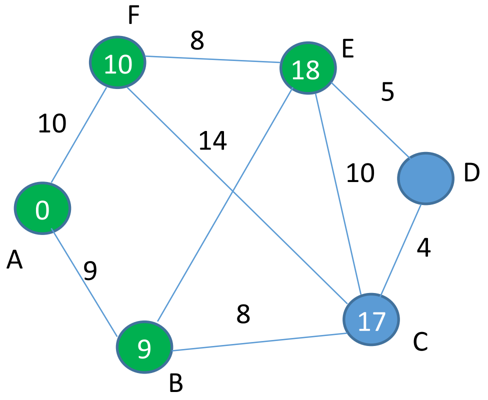
Apskaičiuojame atstumą iki viršūnės E: prie atstumo BE pridedame atstumą AB, užrašome gautą skaičių viršūnėje E.



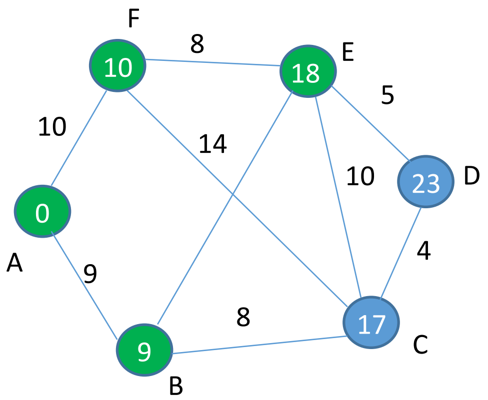
4. Toliau imame viršūnę F ir skaičiuojame atstumus iki jos kaimyninių viršūnių E ir C. Apskaičiuojame atstumą iki E: sudedame atstumą FE ir atstumą AF, palyginame gautą skaičių su skaičiumi, užrašytu viršūnėje E. Užrašome viršūnėje E mažesnįjį skaičių:



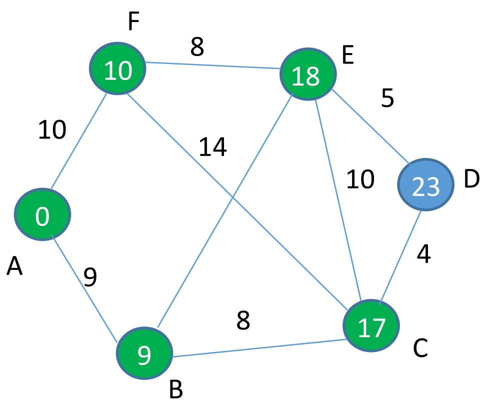
Apskaičiuojame atstumą iki viršūnės C: prie atstumo FC pridedame atstumą AF ir palyginame gautą skaičių su jau įrašytu viršūnės F atstumu. Užrašome (paliekame) mažesnįjį:



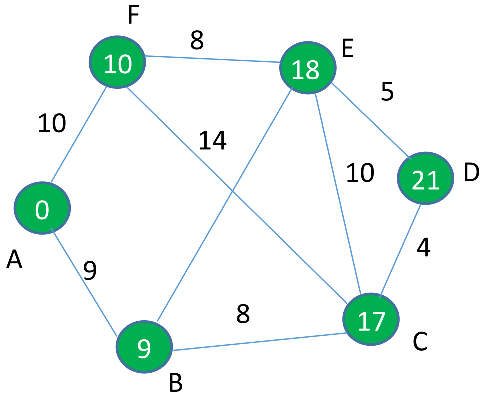
5. Apskaičiuojame atstumus nuo viršūnės E iki jos kaimyninių viršūnių C ir D. Apskaičiuojame atstumą iki D: prie jau suskaičiuoto atstumo iki E pridedame atstumą ED ir užrašome gautą skaičių viršūnėje D.



Apskaičiuojame atstumą iki viršūnės C: prie atstumo iki viršūnės E pridedame atstumą EC, palyginame gautą skaičių su jau įrašytu skaičiumi viršūnėje C. Užrašome (paliekame) mažesnįjį:



6. Apžiūrime viršūnės C kaimyninę viršūnę D. Apskaičiuojame atstumą iki viršūnės D: prie atstumo iki viršūnės C pridedame atstumą CD ir palyginame gautą skaičių su jau užrašytu atstumu iki viršūnės D. Užrašome (paliekame) mažesnįjį:



Gavome trumpiausius atstumus nuo viršūnės A iki visų grafo viršūnių:

AB = 9, AC = 17, AD = 21, AE = 18, AF = 10

Šitokia trumpiausiojo kelio paieška vadinama Dijkstros algoritmu. Jį galima pritaikyti bet kuriam grafui.

Dijkstros algoritmas aprašytas daugelyje šaltinių. „Vikipedijoje“ pateikta animacija, kaip vykdomas Dijkstros algoritmas:

https://lt.wikipedia.org/wiki/Dijkstros_algoritmas

„Bebro“ konkurso uždavinys trumpiausiam keliui rasti

Trys bebrai keliavo iš urvo A į urvą H plaukdami tik pasroviui. Kiekvieno bebro kelias pažymėtas skirtinga linija. Vienas iš bebrų žino Dijkstros algoritmą trumpiausiam keliui rasti:

1. Pirmajam urvui (A) priskiriamas 0. Laikoma, jog A yra „dabartinis“ urvas.

2. Skaičiuojami atstumai nuo „dabartinio“ urvo iki kiekvieno jam gretimo urvo.

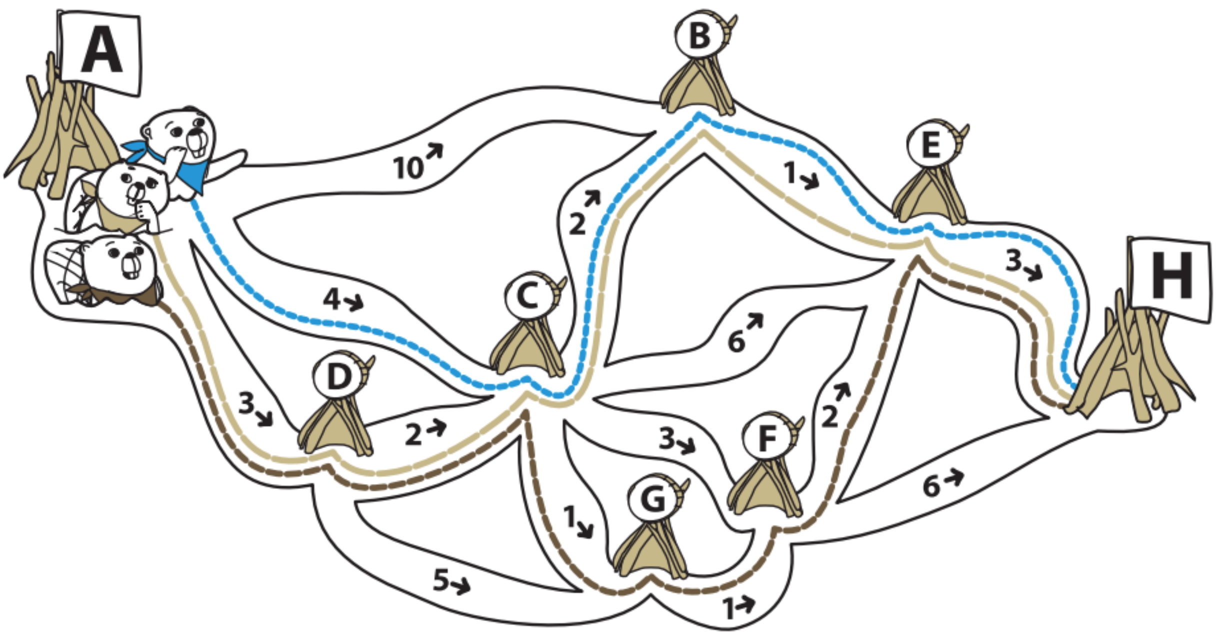
3. Šie atstumai lyginami tarpusavyje ir nustatomas mažiausias skaičius.

4. Mažiausią skaičių pelnęs urvas tampa „dabartiniu“, o prieš tai buvęs urvas pažymimas aplankytu – aplankytas urvas daugiau nebetikrinamas.

5. Grįžtama prie 2-ojo žingsnio.

6. Baigiama, kai pasiekiamas H urvas.

Kuris iš bebrų naudojo Dijkstros algoritmą?



Aptarkite...

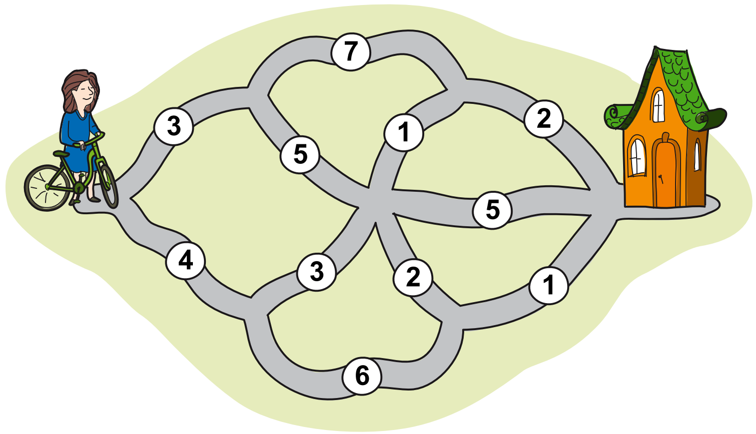
Kaip būtų galima keisti šį uždavinį, kokius klausimus dar galima kelti šiam uždaviniui? Kokio amžiaus mokiniams jis tinkamas?

Trumpiausio kelio radimo uždaviniai 5–6 klasei

Išnagrinėkite pateiktus trumpiausio kelio radimo uždavinių pavyzdžius.

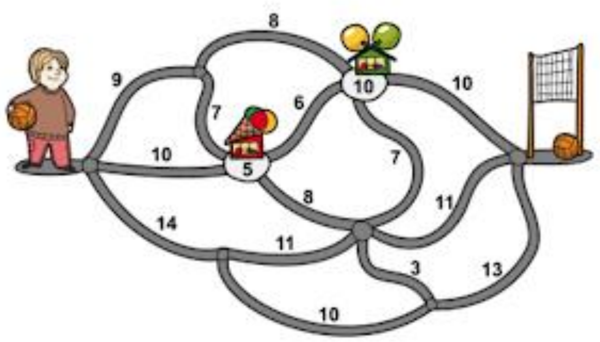
Porose sukurkite du uždavinius, kurių sprendimui būtų galima pritaikyti Dijkstros algoritmą, kad būtų galima vienareikšmiškai ir greitai surasti trumpiausią kelią.

1 uždavinys. Lina nori dviračiu nuvažiuoti į draugės namus. Skaičiai rodo, per kiek mažiausiai minučių Lina gali nuvažiuoti atstumą dviračiu.



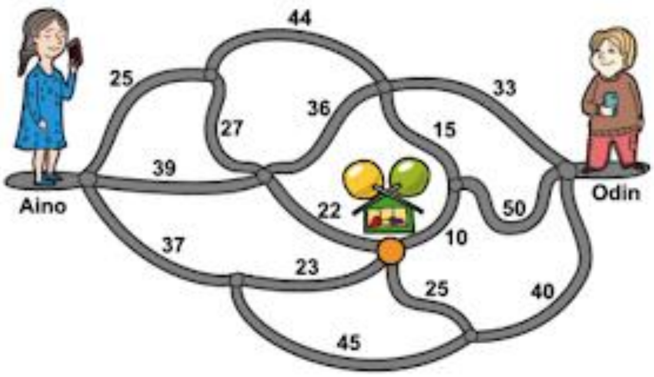
Lina skuba. Raskite trumpiausią (greičiausią) Linos kelią ir apskaičiuokite, kiek laiko (minutėmis) reikės šiai kelionei įveikti.

2 uždavinys. Beno treniruotė prasideda 17 val. Eidamas pro ledainę ar saldainių parduotuvę Benas mėgsta užeiti ir nusipirkti saldumynų. Ledams nusipirkti užtrunka 5 minutes, o saldainiams – 10 minučių.



Kada vėliausiai Benas turi išeiti iš namų, kad laiku atvyktų į treniruotę? Skaičiai rodo, kiek minučių Benui reikia eiti.

3 uždavinys. Benas ir Sigita susitarė susitikti saldainių parduotuvėje. Skaičiai rodo, kiek metrų vaikams reikia nueiti.



Apskaičiuokite, kokiu trumpiausiu keliu kiekvienas vaikas gali nueiti į saldainių parduotuvę.

Iššūkis!

Kiek gatvių reikia norint sujungti *n* namų?

Optimalus sprendimas: reikia *n*–1 gatvių.

Šaltiniai

What is Dijkstra's algorithm. Edukativee.
https://www.educative.io/answers/what-is-dijkstras-algorithm

L. Petrauskas, J. Skūpienė. Informatikos olimpiados: algoritmai ir taikymo pavyzdžiai. Nacionalinė moksleivių akademija. 2006.
https://inf-knyga.nmakademija.lt/lt/latest/11_med%C5%BEiai.html