|  |
| --- |
| Vilniaus universitetas |
| Vilniaus Universitetas |
| Matematikos ir informatikos fakultetas |
|  |
| **Lauryno Paradausko 3 kurso 1 grupės** |
| **Algoritmų analizės 0 namų darbas** |

|  |
| --- |
|  |

Uždavinio formuluotė

* Matematinis apibrėžimas

**Duota:** Mišrus grafas G=(V,A,E), ilgis l(e)N kiekvienai e .

**Ieškomas rezultatas:** Ciklas G grafe (gali turėti pasikartojančių viršūnių), kuris susideda iš vienpusių ir dvipusių briaunų praeinant pro jas tik vieną kartą (keliaujant pro vienpuses briaunas tik pagal jų kryptį).

**Surasti:** Ciklo ilgį (minimizavimo uždavinys).

* + Istorinis

Viskas prasidėjo nuo kinų laiškanešio problemos, kuri yra grafų teorijos nagrinėjama sritis. Tai dar yra žinoma kaip „kelio apžiūros problema“ (angl. Route inspection problem). Pavyzdžiui, laiškanešiui reikia pristatyti paštą į visą apylinkę. Jis yra tingus, taigi jis nori surasti trumpiausią kelią pro apylinkę, kuris atitinka kriterijus:

* Maršrutas prasideda ir užsibaigia tame pačiame taške.
* Reikia apeiti visas gatves bent po kartą.

Problemą pirmasis pradėjo spręsti kinų matematikas Mei-Ku Kuan 1962 metais.

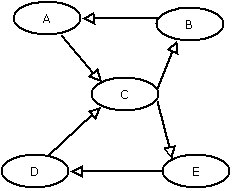
Globalizacijai įsibėgėjus, keitėsi ir problemos tema. Atsiradus mašinoms, transporto kamščiams, keliams su vienpusiu ir dvipusiu eismu atsirado ir problema „New York street sweeper problem“ (New York gatvių valymo problema). Kuomet sankryžos yra taškai, o keliai – briaunos, kai kurios su kryptimi (reiškiančia vienpusio eismo kryptį). Pastarąją problemą ir apžvelgsiu plačiau.

Uždavinio pavyzdys

Paprasčiausias sprendimo būdas – įsitikinti, kad grafas turi Eulerio ciklą. Taigi tokiu atveju ciklo ilgis yra visų briaunų suma.

Pagrindinė taisyklė (kinų laiškanešio atveju):

Eulerio ciklą turi orientuotas grafas tada ir tik tada, jeigu kiekviena viršūnė turi lygių įeinančių ir išeinančių briaunų skaičių.



Visų viršūnės turi vienodą išeinančių ir įeinančių kelių skaičių. Tarkime mūsų pašto skyrius yra C. Pradėdami nuo C galime eiti į B arba E, didelio skirtumo nėra kur eisime, nes vis tiek reikės apeiti visas „gatves“. Einame į E, tada į D ir grįžtame į C. Analogiškai grįžę į pradinį tašką einame į B->A ir į C.

Kas žinoma pasaulyje

Pasaulio matematikų ši problema yra senai žinoma, dažnai sprendžiamas klausimas „Kaip žiemos metu reguliuoti sniego valymo mašinų maršrutą, kad jis būtų geriausiai suplanuotas?“. Taip pat yra Fleury‘s algoritmas, kuris sprendžia šį uždavinį. Problema labai išpopuliarėjo 1965 metais kuomet Edmonds pasakė kalbą apie Guan‘o sprendžiamą klausimą.

Blogiausias atvejis

Brutalios jėgos algoritmo sudėtingumas yra O(N+M), kur N – viršūnių, o M – briaunų skaičius.

Euristinis algoritmas

Iteracinis

Tarkime, kad duotas multigrafas yra jungus, ir visų jo viršūunių u laipsniai yra lyginiai. Pasirinkę bet kurią pradinę e viršūunę, sukonstruosime Oilerio ciklą. Iš viršūunės v0 einame bet kurią grafo briauna (v0; u), iš viršūunėes u vėl bet kuria briauna (u;w) ir t.t., kiekvieną jau panaudotą briauną pašalindami iš briaunų aibės. Kadangi grafo viršūunių laipsniai lyginiai, tai atėjus briauna į bet kurįą grafo viršūunę visada dar turėsime bent vieną briauną išejimui iš šios viršūunės. Galų gale mes pakliūsime į pradinę viršūnę (nes iš visų kitų galėjome išeiti). Jei dar liko briaunų, kuriomis galime išeiti iš šios viršūnės (t.y., jei jos laipsnis pradžioje buvo didesnis už 2), tai vėl einame bet kuria briauna ir vėl kažkada grįšime į v0. Taip tėsiame tol, kol grįžę į v0 iš jos išėiti nebegalime. Gavome ciklą C. Gali būti kad dar liko neapeitų briaunų. Nagrinėsime pradinio grafo G pografį G’, kuris gavosi iš pradinio grafo pašalinus visas ciklui C priklausančias briaunas ir izoliuotas viršūnes (pografis G’ nebūtinai jungus). Šis pografis būtinai turės bent vieną bendrą viršūnę su ciklu C (nes priešingu atveju ciklas C būtų buvės atskira pradinio grafo komponentė, o taip negali būti). Tarkime, kad tai yra viršūnė v1, po to einame likusia ciklo C dalimi, jungiančia viršūnes v1 ir v0:

C := C(v0,v1) U C1 U C(v1, v0).

Taip tęsiant galų gale į ciklą C paklius visos duoto grafo briaunos.