

Mittelineaarsed struktuurid

Graaf. Põhialgoritmid graafidel.

Graaf

Ainult lineaarses vormis ei ole võimalik kõiki andmeobjektide vahelisi seoseid modelleerida.

Graaf on struktuur, millega kujutatakse paari-kaupa seoseid objektide hulgas.

Erijuhul võib olla ka siin tulemuseks lineaarne struktuur. Üldiselt saame aga üsna suvalise kujuga tippudest ja servadest koosneva võrgu

Ajalugu

Esimese graafiteooria probleemi **Königsbergi sildadest** lahendas 1736. a. Šveitsi matemaatik ja füüsik Leonhard Euler (1707-1783)

Euleri tulemused jäid pikaks ajaks unustusse ning graafe on korduvalt „taasavastatud”. Nii avastas 1845. a. need saksa füüsik G. R. Kirchhoff (1824-1887) vooluahelate kohta käivates Kirchhoffi seadustes.

Graafi mõiste võttis 1878. a. kasutusele inglise matemaatik James Joseph Sylvester (1814-1897).

Königsbergi sillad

Königsbergi sildade probleem oli esimene graafiprobleem (ehkki sõna „graaf“ ei kasutatud)

Pregeli jõel Königsbergis oli L. Euleri eluajal 7 silda. Linna elanikel olla olnud mäng, mille käigus püüti leida jalutamisteed.

- Probleem 1: kas lugupeetavad linnakodanikud saavad pühapäeval jalutuskäigul ületada kõik sillad ühe korral ja jõuda koju?
- Probleem 2: kas kodanikud saavad planeerida jalutuskäigu nii, et iga sild ületatakse ühe korra, kuid tagasi lähtepunkti jõudmine ei ole oluline?

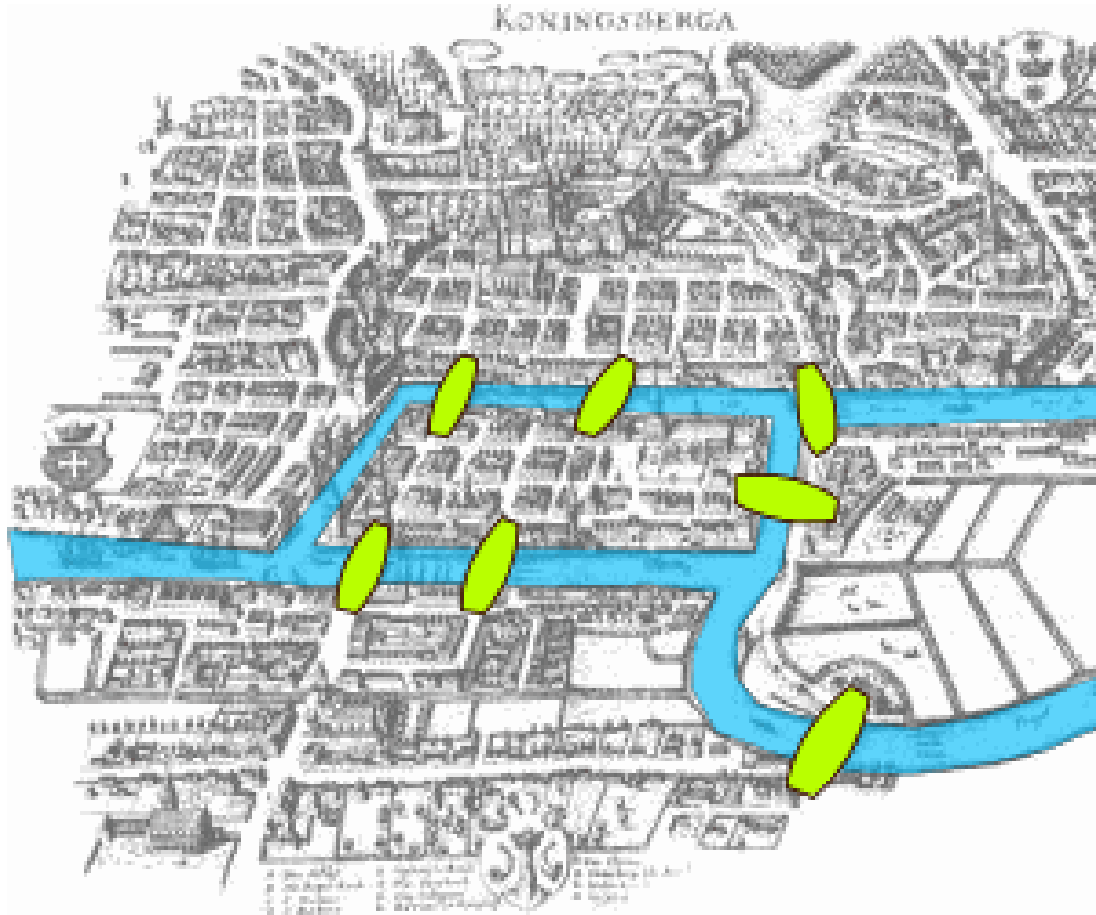
Pildid järgmistel slaididel:

http://et.wikipedia.org/wiki/K%C3%B6nigsbergi_sildade_probleem

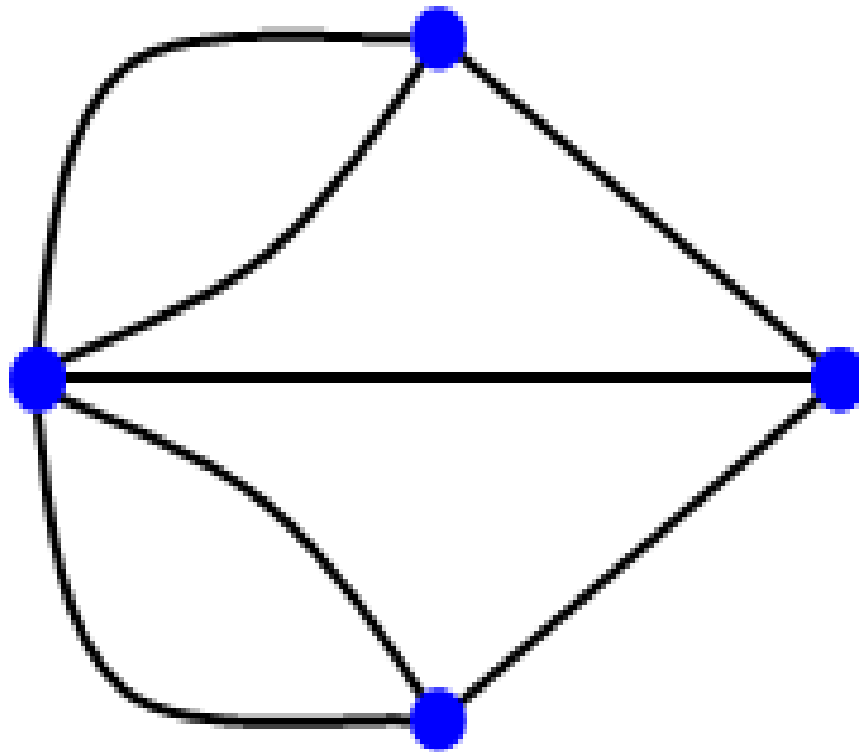
Lugeda saab ka siit:

<https://www.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem>

Königsbergi sillad – kaart (1651. a)



Königsbergi sillad - graaf



Graafi definitsioone

Graaf $G=(V, E)$ on järjestatud paar mittetühjast hulgast V ja selle hulga elementide paaride hulgast E .

Hulga V elemente nimetatakse **graafi tippudeks** ja hulga E elemente **graafi servadeks, kaarteks** või **seosteks**.

Graaf (ingl *graph*) koosneb tippudest (ingl *vertex, node*) ja tippe ühendavatest **kaartest** (servadest) (ingl *edge*).

Graaf

Suunatud (orienteeritud) graaf (ingl *directed graph, digraph*) on graaf, mille kaartel on suund, st iga kaare jaoks on määratud algusetipp ja lõputipp. Öeldakse, et graafi tipud on paari kaupa järjestatud.

Suunamata (orienteerimata) graafis (ingl *undirected graph*) on seos kahe tipu vahel mõlemas suunas ühesugune (seos on sümmeetriline). See kehtib graafi kõigi servade kohta.

Tippude seotus

Tee (ingl *path*) ehk **ahel** on selline kaarte järgnevus, kus ühe kaare lõpp-punkt on järgmise kaare alguspunktiks.

Kaks tippu v ja u on **seotud**, kui nende vahel on tee.

Kaks tippu on seotud **külgnevussuhtega** (ingl *adjacency relation*), kui ühest tipust läheb kaar teise tippu (tipud külgnevad).

Kui kaar läheb tipust v tippu w , siis on tipp w **tipu v naaber** ehk temaga **külgnev tipp**.

Ahelad ja tsüklid

Elementaarahel on ahel (kaarte ehk servade jada), mis ei läbi ühegi tipu üle ühe korra.

Lihtahel on ahel, milles servad ei kordu.

Kui ahela algus ja lõpp on samas tipus, siis on tegemist **tsükliga**

Elementaartsükkel - elementaarahel, mis lõppeb samas tipus

Lihttsükkel - lihtahel, mis lõppeb sama tipus.

Tsüklid

Hamiltoni tsükkel on elementaartsükkel, mis läbib kõiki graafi tippe (ühe korra).

Euleri tsükkel on lihttsükkel, mis läbib kõiki graafi servi (ühe korra).

Sidus graaf on graaf, milles iga kahe tipu jaoks leidub neid tippe ühendav tee. Ehk graaf on piltlikult öeldes ühes tükis

Königsbergi sillad - lahendus

Euleri tee on tee, mis läbib graafi kõik servad ühe korra (probleem 2).

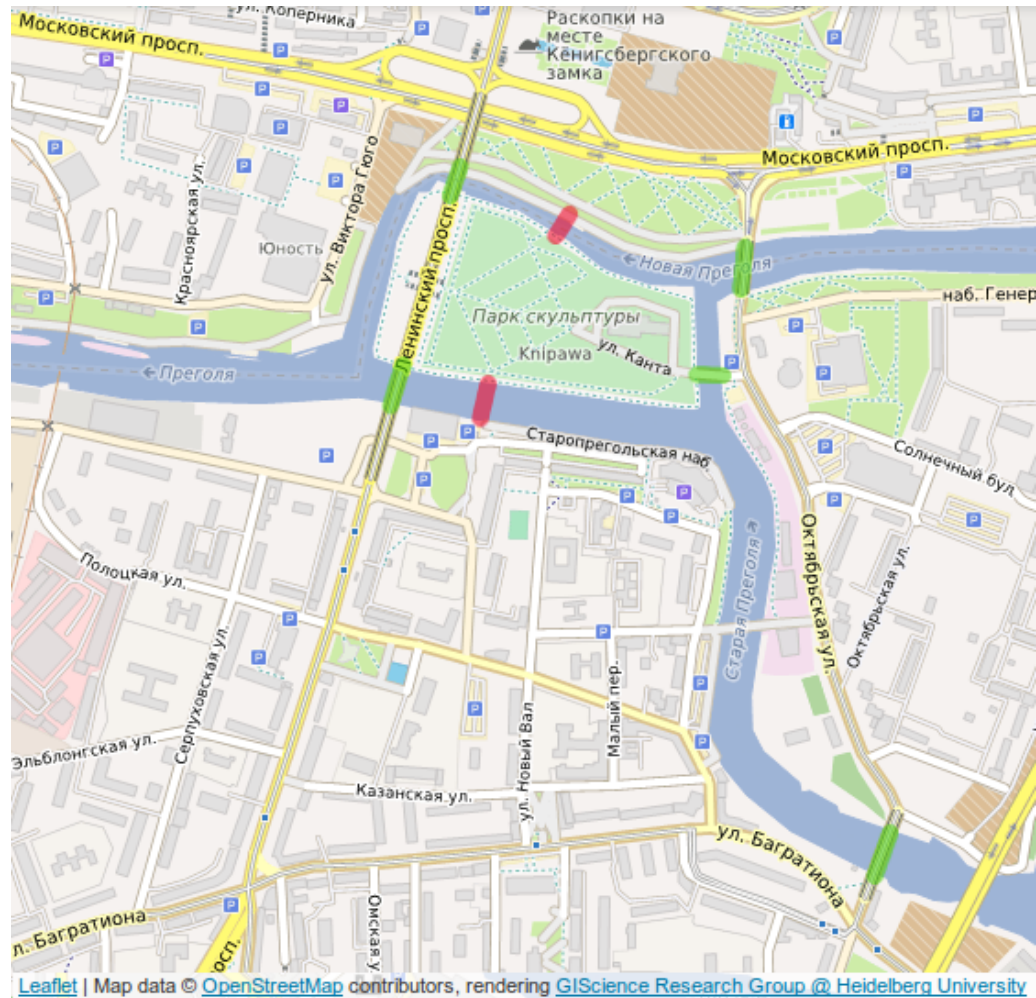
Euleri tsükkel — Euleri tee, mis moodustab tsükli ehk jõuab lähtetippu tagasi, olles eelnevalt läbinud kõik graafi servad ühe korra. (probleem 1).

Tipu aste (valents) on servade arv, mille otspunktiks antud tipp on.

Vastavalt Euleri teoreemile esineb **Euleri tsükkel** graafis siis, kui graaf on sidus ja selles ei ole paaritu arvulise astmega tippe.

Euleri tee esineb graafis siis, kui graaf on sidus ja ei sisalda rohkem kui kaht paaritu arvulise astmega tippu.

Königsbergi sillad täna



By Map data by OpenStreetMap contributors; rendering by GIScience Research Group @ Heidelberg University; produced work by Santacloud - <http://openstreetmap.ru/#map=15/54.7044/20.5175&layer=S>, CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=32759025>

Atsükliline graaf

Kui graafis ei ole ühtegi tsüklit (mitte ühestiki tipust ei leidu teed samasse tippu tagasi), nimetatakse graafi **atsükliliseks** (*acyclic graph*).

Atsükliline suunamata graaf on oma olemuselt puu. Ta ei pruugi olla **juurega puu** (*rooted tree*). Juurega puul on fikseeritud juurtipp.

Suunatud atsükliline graaf (*directed acyclic graph* ehk *DAG*) on suunatud graaf, kus puudub tsükel. Selline graaf on oluline mitmete ülesannete lahendamisel.

Info graafis

Info / objektid paiknevad graafi tippudes.

Infot sisaldavad ka graafis servad, esitades seoseid tippude vahel. Seose tähendus võib olla väga erinev.

Graafi servade ehk kaartega saab siduda arvud.

Neid arve kutsutakse **kaaludeks** ning vastavat graafi **kaalutud graafiks** (*weighted graph*) ehk **võrguks** (*network*).

Kaalude abil saab näidata seoste tugevust, pikkust, läbilaskevõimet vms.

Realisatsioon

Graafide realiseerimisel on oluline salvestada infot iga tipu naabertippude kohta. Selleks on kaks peamist võimalust:

- **Külgnevusahelloend** (*adjacency-list*)
- **Külgnevusmaatriks** (ka **naabrusmaatriks**, **seosmaatriks**) (*adjacency-matrix*)

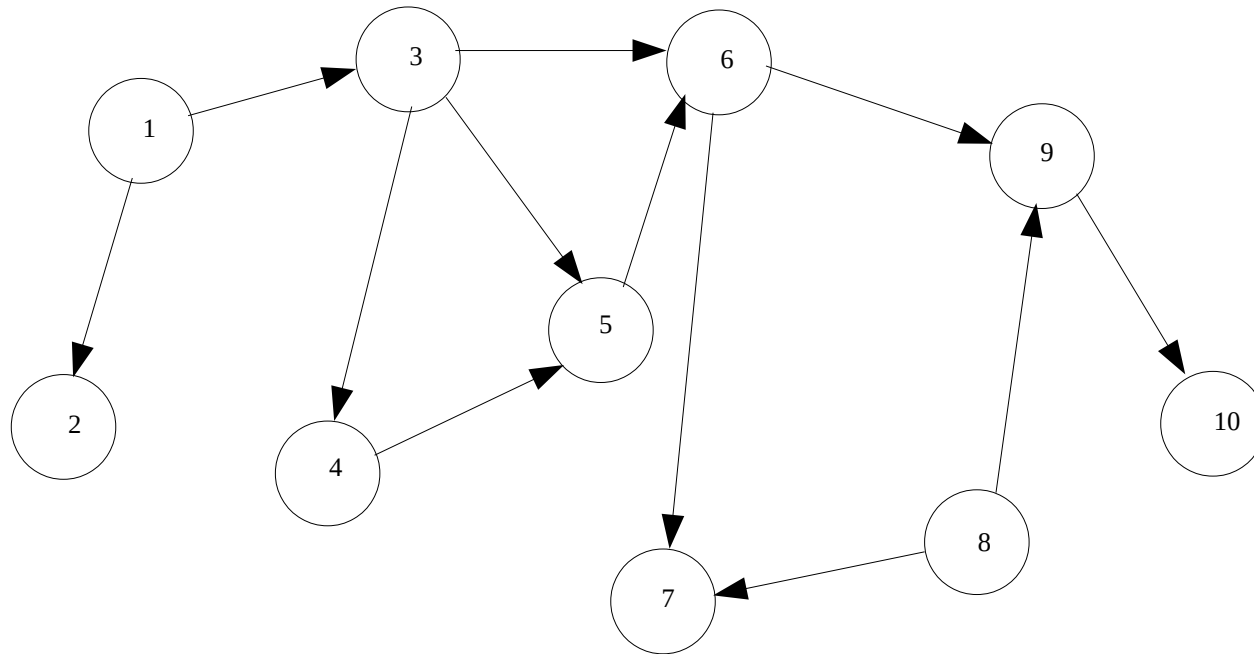
Eelistatav realisatsioon sõltub graafi iseloomust, lahendatavast ülesandest, tippude ja servade suhtarvust jms.

Eristatakse **tihedaid** (*dense*) ja **hõredaid** (*sparse*) graafe, mida iseloomustab erineva tippude ja kaarte arvu suhe. Esimestele on sobivam külgnevusmaatriks (pole liigset mälu raiskamist). Teistele pigem külgnevusahelloend.

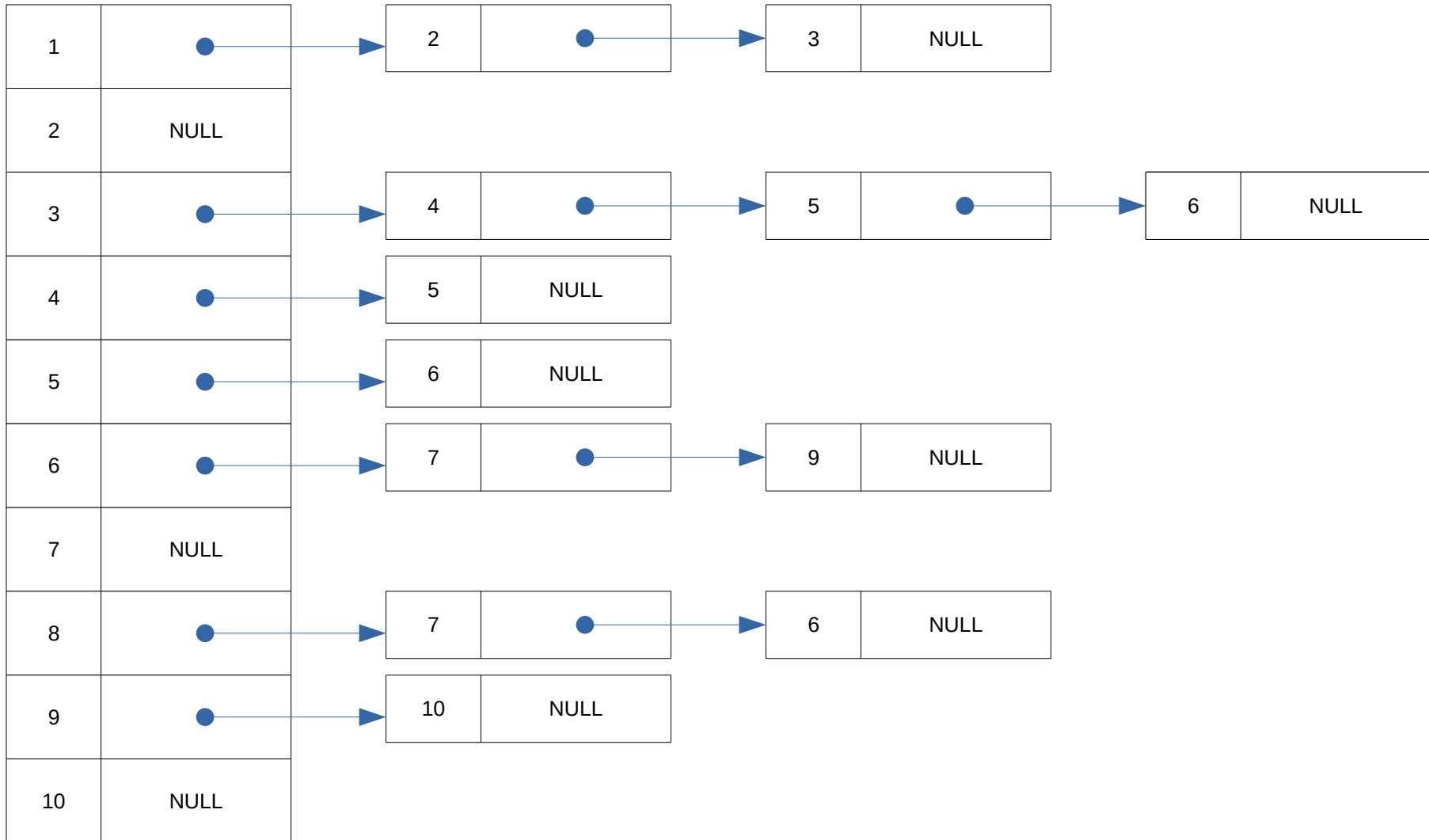
Mõned asjakohased joonised:

<https://de.wikipedia.org/wiki/Adjazenzliste>

Graaf



Külgnevusahelloend



Graafiprobleemid

- Kuidas jõuab kõige kiiremini Tallinnast Värskasse?
- Kuidas transportida kaupa kõige odavamalt erinevatelt müüjatelt erinevatele ostjatele.
- Kuidas korraldada tööde järjekord, et saada maja valmis kõige lühema ajaga?
- Kuidas suunata netiliiklus efektiivselt läbi ruuterite?
- Kuidas joota trükiplaadil kokku kontaktid nii, et kõik oleksid omavahel seotud, kuid seoseid oleks minimaalne arv või et lisatavad “traadid” oleksid minimaalse pikkusega?

Klassikalised graafiprobleemid

On linnade ja nende vaheliste kauguste nimekiri. Milline on lühim võimalik teekond, mis läbib kõik linnad täpselt ühe korra ja jõuab algpunkti tagasi? **Rändkaupmehe ülesanne** (*travelling salesman*)

Naiivse jõumeetodil lahenduse korral, leides kõikvõimalikud teekonnad, on tegemist faktoriaalse keerukusega algoritmiga.

Kui tasapind on jagatud pidevateks piirkondadeks, moodustades kaardi-taolise kujundi, saab selle värvida nelja erineva värviga nii, et naaberpiirkondi ei värvita sama värvi.

Nelja värvi probleem (*four-color problem / theorem*)

Graafialgoritmid

Graafidele on kirjeldatud palju erinevaid algoritme, mis aitavad ülesandeid lahendada.

Kaks olulist üldist graafialgoritmi on **laiutiotsing** ja **süvitsiotsing** (sarnasus vastavate puualgoritmidega).

Mõlema algoritmi sisuks on graafis kõigi tippude läbimine, kuid seda tehakse erinevas järjekorras (tegelikult midagi klassikalises mõttes ei otsita).

Mitmed teised algoritmid kasutavad neid kahte põhialgoritmi.

Laiutiotsing

Laiutiotsing (ingl *breadth-first search*) on strateegia, mille abil on näiteks võimalik leida lühemaid teid.

Laiutiotsingul võetakse aluseks lähtetipp u . Leitakse kõik tipud, kuhu on võimalik jõuda tipust u ühe kaare läbimise järel (tipu u naabrid).

Seejärel võetakse aluseks kõik eelmisel ringil leitud tipud ja leitakse omakorda nende naabrid. Sisuliselt tipud, mis asuvad lähtetipust kahe kaare kaugusel.

Kui tipp oli juba vähema hulga kaarte läbimisega kätte saadud, siis uut pikemat teed ei fikseerita. Nii toimides läbitakse lõpuks kõik tipud ja saadakse teada kaarte arv algustipust igasse tippu.

Laiuti otsimine

Algoritmi töö tulemuseks on loetelu tippudest, mis on kättesaadavad lähtetipust S .

Lisaks on teada tee pikkus (kaarte arv) tipust S kõigisse teistesse tippudesse. Tekib puukujuline struktuur (mida küll füüsiliselt üles ei ehitata), nn **laiutiotsingupuu**.

Näide:

http://en.wikipedia.org/wiki/Breadth-first_search

Süvitsiotsing

Süvitsiotsing (ingl *depth-first search*) strateegia on sarnane labürindi läbimisele, kui seda vähegi süstemaatiliselt teha.

Süvitsiotsingu strateegia seisneb selles, et minnakse ühte teed mööda nii (kaugele) sügavale, kui see võimalik on (st tipult tema naabrile ja sellelt omakorda tema naabrile jne).

Kui tipul rohkem naabreid pole, pöörduakse eelmisesse tippu tagasi ja otsitakse teist teed.

Nii jätkatakse seni, kuni leidub tippe, mida pole veel külastatud, kuid mis on algustipust kättesaadavad.

Topoloogiline sorteerimine

Kui graaf on atsükliline ja orienteeritud (DAG), siis on graafi tippude vahel olemas (ja leitav) **osaline järjestus**.

Topoloogilise sorteerimise ülesanne: leida graafi tippude lineaarne järjestus, kus iga kaar läheb nõ eelneva tipu poolt järgneva tipu poole. Ehk kõigi reas olevate tipupaaride u ja v korral on tipp u eespool tippu v .

Ei ole sorteerimine tavamõistes, kuid annab siiski teatud tippude järgnevuse.

Nt tööde järjestamise (planeerimise) ülesanded.

Topoloogiline sorteerimine

Näide slaidil 5 oleva graafi põhjal.

Eellaste tabel, mille alusel sorteeritud jada moodustatakse:

tipp	1	2	3	4	5	6	7	8	9	10
eellasi	0	1	1	1	2	2	2	0	2	1

Kaks võimalikku järgnevust oleksid:

- 1 2 3 4 5 6 8 9 10 7
- 8 1 3 2 4 5 6 9 7 10

Loe täpsemalt pikast materjalist!!

Toesepuu

Toesepuu (ingl *spanning tree*) on alamgraaf (graafi osa), millesse arvatud servad moodustavad puu, ühendades samal ajal kõik graafi tipud.

Servi on eemaldatud nii, et kaoksid tsüklid, ent sidusus kõigi tippude vahel säiluks. Toes ei ole üldjuhul üheselt määratud.

Kui servadel on mittenegatiivsed kaalud, saab leida **minimaalse toesepuu**. See on toes, mille kogukaal on kõigi toespude hulgast minimaalne.

Näide Kruskali algoritmist:

http://en.wikipedia.org/wiki/Kruskal%27s_algorithm

Dijkstra algoritm

Lühima tee leidmiseks kaalutud graafis saab kasutada Dijkstra algoritmi.

Lühim tee kaalutud graafis on selline tee, kus kaarte kaalude summa on väikseim (mis ei tähenda ilmtingimata vähimat kaarte arvu).

Selle algoritmiga tegeleme edaspidi.