

KODUTÖÖ NR 6. GRAAFIDE ALGORITMID JA NENDE RAKENDUSED

ÜLESANNE 3. DIJKSTRA ALGORITMI TEOREETILINE ANALÜÜS

~~Dijkstra~~ Dijkstra algoritm on lühima tee leidmise algoritm, mida rakendatakse graafide, kus sõlmed on omavahel kas otsest või kaudset ühendatud ning igaal sõlmel teel on ka oma kaal. Lühimat teed lekke võrme vahel loetakse kõige väiksema kaaluuga teeks. Selle leidmisel kasutatakse Dijkstra algoritmi läbi kõike võrmed graafis ning jätab uuesti milline kaalud uinangle leidmise sõlmede vahel olid.

Dijkstra algoritmi ja lühima tee leidmise konsepti saab lihtsalt üle kanda navigeerimis-rakendustesse, mis vastavalt määratud alades ja lõpp-punktile arvutavad kilomeetriažilt kõige optimaalsema reisu. Samaselt saab kasutada Dijkstra algoritmi efektiivselt maasõutimise protokollides arvutivõrkudes.

Kuna Dijkstra ei väärast nimet, sõlmedevahelistes kaaludes mis seda oleks väga ebaspekulatiivne kasutada. Oletades, mis kaale lei sellist ei kasuta, kuid pole vaja. Kui arendada edasi mõtet, unistamisrakendust ja kuna või kasutada sellist linnas, mis on väga märgne või tähtsad on mitte ainult lammutatult mõõda maapiinda vaid ka näiteks suuremale elitise vahel mis tasulise algoritmi väris ei sobiks. Nõ 3D linnas pulul peales arvutama ka kõrguste vahedega ning lastumisel võib leitud unna justkui unneusse. Negatiivsed arvud Dijkstraas aga kehtaks olemasolev levi viima ke leidmisel kehtaks ke unneunneunne. Väärtusena ja kehtaks kehteks vale / mitte kõige lähim ke.

ÜLESANNE 4. BELLMAN-FORDI ALGORITMI TEOREETILINE ANALÜÜS

Bellman-Fordi algoritm on sarnane Dijkstra algoritmile, mõeldud graafis sõlmedevaheliste ühenduste olemasolul lähima ke leidmisel. Üks oluline erinevus nende kahe algoritmi vahel seisub selles, et Bellman-Fordi algoritmis arvestatakse ka juhtudega, kus sõlmede vahel on negatiivsed arvud. Lisaks ei pea sellel graafil ilmtingimata kaole sõlmede vahel olema.

Bellman-Fordi algoritm suudab kuvastada, kas graafis on negatiivse tsükli ning selles käitakse kogu graafi sõlmed läbi, jätkatakse lahkunud punktist A punkti F mulde ning alustatakse uuesti mõnest kindist punktist punktist nt E-st D-st unneunne. Kui selle kõiges tuleb välja,

et laugused on erioluist eluavad siis järeltuleb esineb graafis negatiivne hülkel. Bellman - Fordi algoritm negatiivse hülkel leidmist kasutataksegi ~~et~~ olukordades, kus teadlase sõlme vahel võib eksistada mingi põlgus negatiivses, mingi vältimatuks kummalise tulemuse liikimise kude leidmist khib, seda negatiivset aihu kwardada (ja selle kasutamist lahundada).

ÜLESANNE 5. GRAAFIDE VÄRVIMISE PROBLEEM

Siiavi punitum kokku sõlmede värvimisega puuare - unusta puu andmestruktuuris. Seal oli igal liudlat värvil sõlmed omad veegid, ja omadused. Naaber-sõlme värv oli oluline, saamuti ka see, miku sõlme mingit värvil sõlme all võis olla. Graafide värvimisel lehtivad sama moodi veegid, leid khib olukord, kus neid värv võiks anda sõlmedele mitut moodi mingi sellest tulemusest võivad ka sõlmede enda veegid / vaigulus viineda. Mulus peab ka pidama, et kas kõike või mõni värv ei olei alla kōvuti kire sama värviga. Atvutikaduses soabkei värv kasutada just andmestruktuuride puhul, kus värvide kasutamisel aitab leua tõhusama algoritm. Val saab kasutada graafide värvimist, et muuta tõhusaks kaadiojaamade sageduse jaotamist, uortada ajagraafideid leui mingitel elementidel ei ole samal ajal võimalik kōvraga mingit operatsiooni sooritada.

BOONUS ÜLESANNE

P. VS. NP PROBLEMI

ÜLEVAADE

P. vs. NP. probleemid on konkreetne arvuti-
teaduse lahendamata küsimus - kas
iga probleem, mille lahendust saab kiiresti
kontrollida, on ka kiiresti lahendatav.

P. - polünoomie, algsuutmi, mis suudab leida
lahenduse „küüsti“ ja ~~valitud~~ ~~valitud~~ selle
lahenduse leidmisel. ~~valitud~~ ~~valitud~~ kulus aeg
varieerub polünoomifunktsioonina algsuutmi
nõude suurusst. (kui see lahendada).

ND - ~~määratamata~~ / mitte deterministlike (päris õigest
kemiini eesti keeles ei leidnud), algselt mi saadud
lakundust on lihtne pöördumise ajaga kontrollida
(kuigi kontrollida).

Reaktion auf Stress ist \neq Wern auf psychisches
Nervenzusammenbruch milder

Praegu ei ole mudelid tõestada, et $P = NP$ ehk
 kas lihtsasti lahendatavaid probleeme on ka
 lihtne kontrollida. Juhutiteaduses on see teema
 oluline, sest see aitaks paremini defineerida,
 mis teeb vastavalt lahendatava probleemi algoritmis
 keeruliseks. Kui $P = NP$ tähtsustaks see seda, et
 algoritmideks, mis enne olid keerulised saaks
 leha lihtsamad versioonid. Saaks kasutada
 leini, ehk vähem aja- ja muutmäluaid algoritme.
 Praegu kasutatakse NP - tüüpi algoritme
 kuvakeerisimulatsioonis, olulistes olukordades. Kui $P = NP$
 siis tähtsustaks see seda, et vähesel hulgal lahendusi
 enam ei lootaks. (põhinevad algsel määral, et

$P \neq NP$ ühig. vastand. selle suaduse ära) Näib, et
arvimeetiline kriitika ei loimiks -
võimalava võtme ja kuvalist. tagavad
algoritmid. enam ei läidaks oma esmaüki.
Sama, lehtlik. ka. rühmitiline võtme. ajaveerimise
algoritmide. kohta, võtme saaks. lihtsalt. koda
vastades. veel. lihtsamat. algoritmi.

Kuna. tegu. on. arvutiteaduse. leia. tähtsana.
probleemiga. oles. selle. lahendamise. tagajärjed.
koostaga. nii. positiivsed. kui. ka. negatiivsed.
Oluline. oles. selle. lahendamise. kuu. vähe. male.
uurimistega +. vähe. ma. aja. ja. muu. keerukusega
võidaks. soovutada. võtme. ühig. re. kaadus.
pikemas. perspektiivis. üles. selle, et. ühigid.
lahendused. tuleks. ümber. leha. Samas. kui.
 $P = NP$. tagajärjel. peaks. kriitika. ümber.
teema. avaks. re. lihtlasti. võtme. ühig. - kas.
süs. uue. lahenduse, tööõhimeetete. või. uue.
ühikumi. probleemide.