

Tietorakenteet ja algoritmit

harjoitustyö

Jeremias Körkkö

2586810

Ratkaisu

Tehtävässä etsitään annetusta graafista reittiä, jonka korkein kohta on matalin. Solmut esittävät kaupunkeja ja sivut teitä kaupunkien välillä. Sivujen painotus tarkoittaa tien korkeinta kohtaa.

Reitin löytämiseksi ensimmäisenä muodostan annetun graafin teistä painotusjärjestyksessä muodostetun listan pienimmästä suurimpaan. Tällä pohjustetaan Kruskalin algoritmia.

Seuraavaksi etsin graafin minimivirittävän puun käyttäen Kruskalin algoritmia.

Minimivirittävä puu tallennetaan kaksiulotteiseen listaan seuraavasti: kukin indeksi kuvaa yhtä kaupunkia. Ensimmäinen(=nollas) indeksi kuvaa kaupunkia yksi, toinen indeksi kaupunkia kaksi jne. Mikäli indeksissä eli kaupungin sisällä on monikko, sieltä lähtee tie. Monikon ensimmäinen luku kertoo tien päätepisteen ja toinen tien korkeuden. Koska graafi on suuntaamaton, tiestä luodaan monikko sekä tien lähtö- että päätekaupunkiin(=indeksiin).

Viimeiseksi etsin minimivirittävästä puusta reitin pisteestä yksi päätepisteeseen syvyyshaulla. Pidän kuljetun reitin ja niiden korkeudet muistissa listassa. Kun haluttu reitti löytyy, ohjelma tulostaa kuljetun reitin ja antaa vastauksena matkalla olleen korkeimman kohdan.

Aikakompleksisuus

V=kaupungit, E=tiet

(1) Ohjelman vaativin vaihe on minimivirittävän puun muodostaminen Kruskalin algoritmilla. Algoritmin aikakompleksisuus on **$O(E \log V)$** .

(2) Syötteen luku ja järjestäminen on kompleksisuusluokkaa **$O(E)$** . Kaupunkien määrän kasvaessa syötteen määrä kasvaa nopeasti ja tämä otetaan huomioon.

(3) Viimeinen vaiheen eli syvyyshaun kompleksisuus on **$O(E_{\text{mst}} + V)$** , jossa $E_{\text{mst}} =$ minimivirittävän puun tiet = $1-V$. Eli siis voidaan ilmoittaa muodossa **$O(2V)$** . Nopea toiminto, ja kaupunkien määrän kasvaessa hyvin marginaalinen vaikutus, voidaan sivuuttaa.

Ohjelman aikakompleksisuus on siis $(1) + (2) = O(E + E \log V)$.

Suoritusajat

Arvioidaan tämän perusteella syötteen koko, jonka algoritmi pystyisi käsittelemään 15 sekunnissa:

Alla tilastoja viidestä suurimmasta annetusta testitiedostosta.

kaupunkeja	teitä	suoritus aika(s)	operaatiot $O(E + E \log V)$
500	39 629	0.078	146 586
750	90 782	0.187	351 785
1000	161 124	0.322	644 496
1500	367 055	0.706	1 532 855
2000	653 175	1.203	2 809 325

Taulukosta nähdään että, kun kaupunkien lukumäärä tuplaantuu, teiden lukumäärä n. nelinkertaistuu. Sama pätee myös operaatioille ja sitä kautta suoritusajalle.

Lasketaan kuinka monta operaatiota keskimääräisesti algoritmi suorittaa sekunnissa ja otetaan niistä keskiarvo.

Lasketaan $\frac{\text{operaatiot}}{\text{suoritus aika}}$ jokaiselle taulukon riville ja otetaan niistä keskiarvo.

Keskiarvoksi saadaan 2 053 700 operaatiota/s

Joten viidessätoista sekunnissa pystytään:

$15 \text{ s} * 2\,053\,700 = 30\,805\,500$ operaatioon.

Jatketaan taulukkoa teoreettisilla arvoilla (kaupunkien kaksinkertaistuessa, muut nelinkertaistuvat), kunnes ylitämme 30 805 500 operaation ja 15 sekunnin rajan.

kaupunkeja	teitä	suoritus aika(s)	operaatiot $O(E + E \log V)$
500	39 629	0.078	146 586
750	90 782	0.187	351 785
1000	161 124	0.322	644 496
1500	367 055	0.706	1 532 855
2000	653 175	1.203	2 809 325
4000	2 612 700	4.812	12 023 802
8000	10 450 800	19.248	44 949 200

8000 kaupungilla operaatiot ja suoritus aika menevät huomattavasti yli vertailuarvon, $44\,949\,200 \gg 30\,805\,500$. Eli vastaus siis löytyy 4000 ja 8000 välistä. Kokeillaan 6000 kaupungilla, eli 0.5 kertaisella korotuksella.

Nyt $E = 2\,612\,700 * 0.5 * 4 = 5\,225\,400$ ja $V = 4000 + 0.5 * 4000 = 6000$

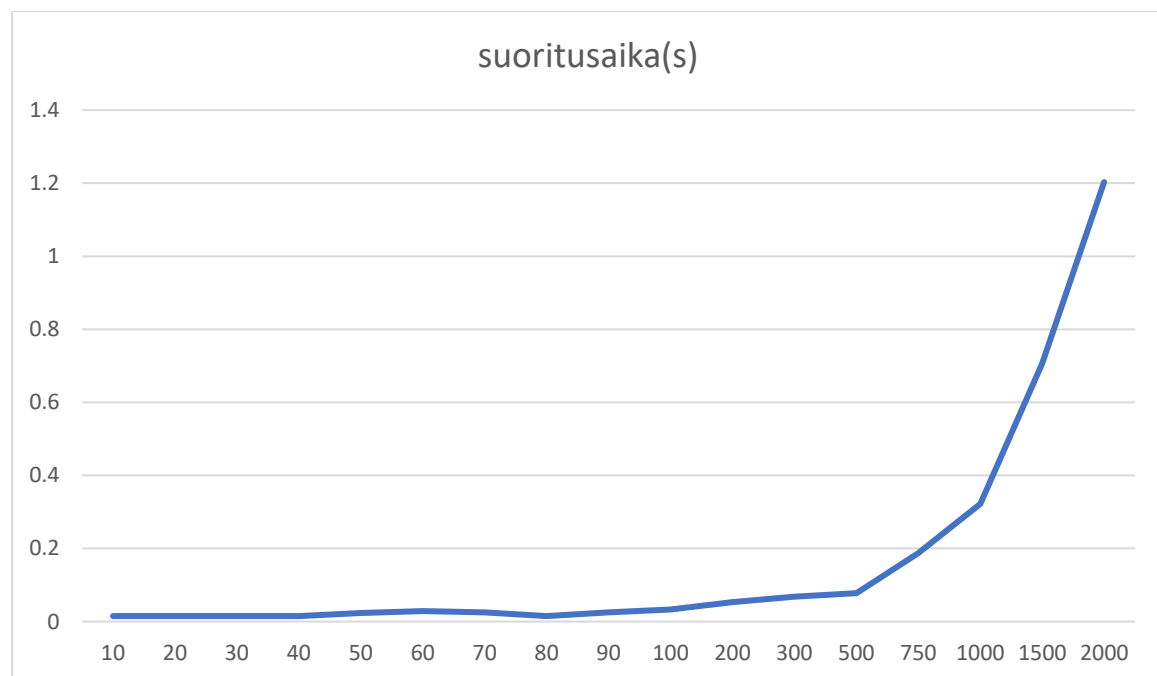
$5\,225\,400 + 5\,225\,400 * \log(6000) = 24\,967\,751 < 30\,805\,500$. Kokeillaan 7000 kaupungilla, eli 0.75 kertaisella korotuksella.

$E = 2\,612\,700 * 0.75 * 4 = 7\,838\,100$ ja $V = 4000 + 4000 * 0.75 = 7000$

$7\,838\,100 + 7\,838\,100 * \log(7000) = 37\,976\,362 > 30\,805\,500$.

Arvo on siis 6000 ja 7000 kaupungin karkeasti arvioituna puolessa välissä. Tämän perusteella arvioin, että **ohjelma pystyy 15 sekunnissa noin 6500 kaupungin kokoiseen verkkoon**.

Testitiedostoilla suoritusaikaa mitatessa saadut arvot kuvaajassa. Vaaka-akselilla kaupunkien lukumäärä ja pystyakselilla suoritukseen mennyt aika sekunteina.



Alle 100 kaupungin verkot ovat pieniä ja niistä ohjelma suoriutuu nopeasti. Sadan kaupungin jälkeen verkot alkavat laajenemaan nopeasti ja ohjelman suoritus sitä myötä hidastumaan. Kaikista annetun kokoisista testitiedostoista algoritmi kuitenkin suoriutui kohtuullisessa ajassa.