Jaottomien polynomien löytäminen äärellisten kuntien yli - Testausdokumentti

Aineopintojen harjoitustyö: Tietorakenteet ja algoritmit

Sebastian Björkqvist

7. syyskuuta 2014

Yksikkötestit

Jokaisella ohjelman tärkeällä algoritmilla ja tietorakenteella on yksikkötestit. Hankalinta on Rabinin jaottomuustestin testaaminen, koska jaottomien polynomien löytäminen käsin on hyvin hankalaa ja aikaavievää. Tällaisia testejä on kuitenkin tehty muutama.

Yksikkötestit voi suorittaa joko ajamalla juurikansiossa sijaitseva skripti run_tests.sh, tai komennolla ant test projektikansiossa Tiralabra.

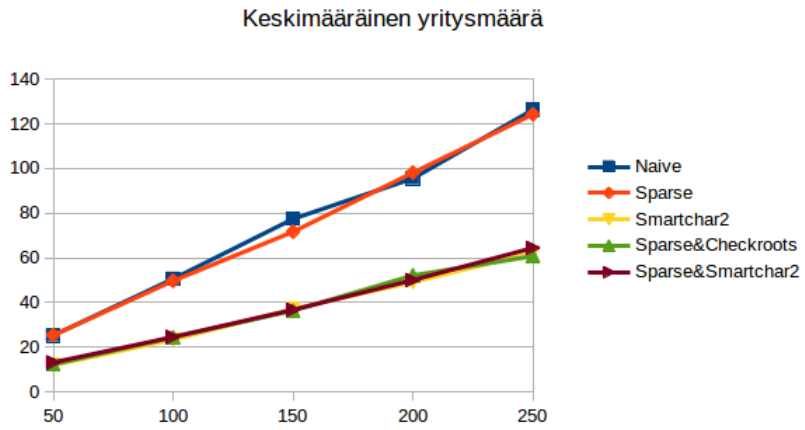
Heuristiikkojen testaaminen

Testattujen polynomien määrä

Eri heuristiikkojen tehokkutta testattiin generoimalla 1000 jaotonta polynomia karakteristikalla 2 ja asteilla 50, 100, 150, 200 ja 250. Taulukossa 1 ja kuvassa 1 näkyy keskimääräinen määrä Rabinin algoritmille annettuja polynomeja.

Aste	naive	sparse	smartchar2	sparse_checkroots	sparse_smartchar2
50	25.3	25.4	12	12.3	13.2
100	50.6	49.7	23.4	24.5	24.6
150	77.5	71.7	37.1	36.4	36.8
200	95.6	98.2	49.2	52.2	50.1
250	126.4	124.3	62.3	60.8	64.5

Taulukko 1: Keskimääräinen määrä eri heuristiikkojen Rabinin algoritmilla testattuja polynomeja



Kuva 1: Keskimääräinen määrä eri heuristiikkojen Rabinin algoritmilla testattuja polynomeja

Tuloksista nähdään, että naiivi algoritmi joutuu tekemään melko tasan $d/2$ yritystä löytääkseen astetta d olevan jaottoman polynomin, kun taas algoritmit checkroots ja smartchar2 selviävät $d/4$ yrityksellä. Tämä on linjassa sen kanssa mitä toteutusdokumentissa todettiin. Koska karakteristika on 2 tekevät algoritmit checkroots ja smartchar2 saman asian, mutta smartchar2 tekee sen nopeammin, koska sen ei tarvitse evaluoida polynomeja.

Lisäksi huomataan että algoritmin sparse käyttö ei juurikaan vaikuta kokeiltavien polynomien määrään. Täten voidaan todeta että sparse-generointi ei ainakaan testatuissa tapauksissa muuttanut valittujen polynomien jakaumaa niin paljon että se vaikuttaisi jaottomien polynomien esiintymistiheyteen.

Tätä suurempia karakteristikoita tai asteita ei testattu, koska asteen 250 testien ajaminen kestää jo lähes vuorokauden. Jos testit kuitenkin halutaan toistaa, löytyy kansiosta test_input_files tiedostot 2_ASTE_large.txt (missä ASTE korvataan halutulla asteella 50, 100, 150, 200 tai 250) jotka voi antaa syötteenä ohjelmalle halutun heuristiikan kera. Testien tulosten raakadata löytyy kansion test_results alakansiosta largerun.

Naive vs. sparse-generointi

Naive- ja sparse-generointeja testattiin mittaamalla 1000 jaottoman karakteristikaa 2 olevan polynomin löytämiseen käytettyä aikaa asteilla 50, 100, 150, 200 ja 250. Tulokset näkyvät taulukossa 2. Saman asteen generoinnit ajettiin samalla ukko-klusterin koneella peräkkäin, mutta tuloksiin ei voi kuitenkaan tuijottaa liikaa, sillä muu kuorma käytetyllä koneella on saattanut vaikut-

taa ajanoton tulokseen. Vaikuttaa kuitenkin siltä, että sparse-generointi on muutaman prosentin nopeampi kuin naiivi generointi.

Aste	naive	sparse	Ero prosenteissa
50	32	30	-6.67
100	588	548	-7.30
150	3613	3569	-1.23
200	14031	14030	-0.01
250	53486	51886	-3.08

Taulukko 2: 1000 jaottoman, karakteristikaa 2 olevan jaottoman polynomin laskemiseen käytetty aika sekunneissa naive- ja sparse-generoinnilla.

Kuten taulukosta näkyy kesti näidenkin testien ajo yhteensä vuorokauden verran. Testien vaatimat input-tiedostot ovat samat kuin aikaisemmassa testissä: Kansion test_input_files tiedostot 2_ASTE_large.txt.