

Deadline 1

Tiedostolukija\kirjoittajaa testattu lukemalla tiedosto ja välittömästi kirjoittamalla se takaisin uuteen tiedostoon ja vertailemalla silmä määräisesti tekstiä. Näyttäisi toimivan oikein, pitänee kirjoittaa yksikkötestit. Ison tiedoston lukeminen\kirjoittaminen hidasta, n. 7 sekuntia 3 megan tiedoston lukemiseen ja kirjoittamiseen. Vaihtamalla Data(Input\Output)Stream vastaaviksi BufferedStreameiksi suoritussyky parani merkittävästi, n. Sekunti tiedoston lukemiseen. Jatketaan tällä linjalla.

OmaArrayListille ja OmaMinimiPriorityQueueulle kirjoitettu yksikkötestit ja varmistettu että läpäisevät testit.

Deadline 2

Algoritmia testailtu käsin, vertailtu alkutiedoston ja puretun tiedoston hasheja ja todettu ne samoiksi jolloin tiedostot suurella todennäköisyydellä on sama. Toteutettu hashmap ja tälle yksikkötestit

Deadline 3

Algoritmia edelleen testailtu käsipelin muutosten jälkeen vertailemalla hasheja. Bugien metsästyks välillä ollut aikaavievää joten älykkäämpi ihminen olisi varmaan tehnyt yksikkötestit jolloin bugin paikallistaminen olisi ollut helpompaa - toisaalta suuret muutokset algoritmin toteutukseen ja osittain itse algoritmiin (koodien kanonisoinnin lisäys esim) rikkoo yksikkötestit joten en osaa sanoa olisiko yksikkötestien jatkuviin päivityksiin lopulta mennyt enemmän aikaa.

Kaikille luokille nyt yksikkötestejä; kattavuutta voinee vielä parantaa varsinkin väärin syötteiden osalta. Pohdin vielä josko luokkia pitäisi pilkkoa pienemmiksi yksityisten metodien testaamiseksi.

Deadline 4

Yksikkötestejä on laajennettu hieman, mutta en ole edelleenkaan tyytyväinen näiden kattavuuteen. Ennen palautusta tätä on parannettava. Tein myös nopeustestailua sekä omien tietorakennelämentaatiodien ja algoritmin kanssa. Prioriteettijonon tehokkuus (tai tehottomuus) pistää silmään; javan oma on ~8 kertaa nopeampi. Hieman ihmetyttää se, että prioriteettijono nopeutui huomattavasti kun sisäisesti omaa ArrayListiä ei enää käytettykään rajapinnan kautta vaan suoraan; noin 30% nopeutuminen. Ylimääräinen muistiluku ilmeisesti maksoi huomattavasti kun ajon aikana jouduttiin tarkistamaan mikä luokka oikeasti asuikaan rajapinnan takaan.