Deadline 1

Tiedostolukija\kirjoittajaa testattu lukemalla tiedosto ja välittömästi kirjoittamalla se takaisin uuteen tiedostoon ja vertailemalla silmämääräisesti 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 suorityskyky parani merkittävästi, n. Sekunti tieodoston lukemiseen. Jatketaan tällä linjalla.

OmaArrayListille ja OmaMinimiPriorityQueuelle 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 metsästys 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 väärien syötteiden ostalta. Pohdin vielä josko luokkia pitäisi pilkkoa pienemmiksi yksityisten metodien testaamiseksi.

Deadline 4

Yksikkötestejä on laajennettu hieman, mutta en ole edelleenkään tyytyväinen näiden kattavuuteen. Ennen palautusta tätä on parannettava. Tein myös nopeustestailua sekä omien tietorakenneimplementaatioiden 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.

Deadline 5

Ohjelma on sellaisessa vaiheessa että olen varsin vakuuttunut että se toimii oikein. Olen keskittynyt lähinnä yksikkötestien laajentamiseen sekä nopeusvertailujen viimeistelyyn. Yksikkötestejä varten pilkoin muutaman luokan taas eri luokiksi jotta voin testata sellaisia metodeja suoraan jotka ennen olivat yksityisiä. Testien määrä per luokka on edelleen aika matala, mutta luokilla tuppaa olemaan vain 2 -3 metodia joista yleensä vain yksi julkinen joten en ole varma miten testejä edes voi helposti laajentaa.