Tietorakenteet ja Algoritmit Harjoitustyön työselostus

18.8.2017

Samuli Helttunen

2307248

1. **Ratkaisu**

Harjoitustyön tavoitteena oli tehdä toimiva ohjelma, joka lukee tekstitiedostoja, tallentaa ne, laskee tiedostoissa esiintyvvien fraasien esiintymismäärät ja lopulta tulostaa 100 eniten esiintyvää fraasia. Tässä harjoitustyössä käytin ohjelmointikielenä C:tä.

Ensin alustin main-tiedoston lukemaan tehtävässä käytettyjä tekstitiedostoja. Ohjelma pyytää käyttäjää syöttämään lukuarvon 1:n ja 3:n väliltä. 1 avaa small.txt:n, 2 medium.txt:n ja 3 big.txt:n. Jos käyttäjä syöttää jotain aivan muuta, hänelle annetaan virheilmoitus. Seuraavaksi aloin työstämään tapaa tallentaa tekstitiedostojen fraasit. Päätyin käyttämään hash-taulukkoa. Hash-taulujen ja hash-funktioiden käytöstä löytyi Internetistä paljon dokumentointia ja esimerkkejä, joiden avulla pystyi itse luomaan harjoitustyön tavoitteen suorittavan ohjelman. Traharkka.c-tiedostosta löytyy kaikki hash-taulukkoihin tarvittavat funktiot, joita voidaan sitten käyttää main-tiedostossa. Jotta funktioita voidaan main-tiedostossa käyttää, ne pitää mainita header-tiedostossa. DictCreate() luo hashtaulun kun taas DictAdd() ja DictInsert() lisää sinne fraaseja. DictDelete() poistaa tallennettuja fraaseja, mutta sitä ei tässä tehtävässä käytetty ollenkaan. DictDestroy() tuhoaa hashtaulun. Tämä funktio on tärkeä grow()-funktiossa.

Kun hash-taulun alkuperäinen tallennuskapasiteetti ylitetään, grow() luo väliaikaisen kopion luodusta taulusta ja luomishetkenä nostaa kapasiteettia GROWTH\_FACTOR:n arvon mukaisesti. Tämän jälkeen väliaikainen taulukko vaihdetaan alkuperäisen kanssa ja vanha taulukko tuhotaan. Grow() aiheutti hieman hankaluuksia joidenkin tiedostojen kanssa, joten nostin INITIAL\_SIZE:n arvoa. Näiden jälkeen lisäsin main-tiedostoon luettujen tekstitiedostojen fraasien tallentamisen.

Seuraavaksi ohjelman piti löytää useimmin esiintyvän fraasin. Erikoisesti nimetty FindLargest() tarkistaa jokaisen taulukon jäsenen kerrallaan ja tarvittaessa vaihtaa largest-pointerin arvoa. Jotta funktiolla voitaisiin saada esimerkiksi 100 useiten esiintyvää fraasia, funktion asettaa ennen palauttamista useimmin esiintyvän fraasin esiintymismäärän arvon negatiiviseksi, tehden taulukon toiseksi useimmin esiintyvän fraasin useimmin esiintyväksi. Tällöin FindLargest()-funktiota voidaan käyttää helposti main-tiedostossa 100 kertaa for-loopin avulla.

1. **Käytetyt tietorakenteet**

Tässä työharjoituksessa on käytetty hash-taulukkoa ja hash-funktioita.

1. **Ohjelman suorituskyky**

Ohjelman suorituskykyyn vaikuttaa käsiteltävän tiedoston koko. Tiedosto lukee ja tallentaa tekstitiedoston fraasit rivi riviltä (Tallentamiseen tarvittavissa funktioissa yhteensä 37-riviä koodia, suurin ohjelman toiminto), joten suurimman tiedoston big.txt:n kanssa ohjelman suoritusaika on noin 4 sekuntia.