811312A Tietorakenteet ja algoritmit 2017-2018, C-kielinen harjoitustyö

Tässä työssä etsitään suuresta tekstitiedostosta sen 100 yleisimmin esiintyvää sanaa. Ohjelma toteutetaan C-kielellä. Sanaksi katsotaan yhtenäinen kirjainmerkkien a..z ja A..Z jono. Lisäksi sanaan voi kuulua vielä heittomerkki ’. Halutessasi voit sisällyttää kirjaimiin myös skandit å,ä ja ö sekä Å,Ä ja Ö, mutta tämä ominaisuus ei ole pakollinen. Isot ja pienet kirjaimet samaistetaan. Esimerkiksi tekstissä

Herman Melville’s book Moby Dick starts, as we all know, with the sentence ”Call me Ishmael”.

sanat olisivat

herman, melville’s, book, moby, dick, starts, as, we, all, know, with, the, sentence, call, me **ja** ishmael

Tekstitiedoston nimi annetaan ohjelmalle joko komentoriviparametrina tai käyttäjän syötteenä. Ohjelma tulostaa tiedoston 100 yleisintä sanaa ja niiden esiintymiskerrat tiedostossa. Sanat tulostetaan esiintymiskertojen mukaisessa järjestyksessä, yleisin ensin.

Koska tiedosto voi olla erittäin suuri, tarvitset sopivan tietorakenteen tallentamaan sanoja ja niiden esiintymiskertoja. Tällainen voi olla esimerkiksi hash-taulu tai binäärinen etsintäpuu. Kaikkia sanoja ei myöskään tarvitse järjestää esiintymiskertojen suhteen, joten 100 yleisimmän sanan hakemiseksi voi käyttää vaikkapa maksimikekoa. Järjestäminen jollakin nopealla lajittelualgoritmilla on myös toki mahdollinen.

Arvioi ohjelmasi aikakompleksisuutta, kun syötteen koon mittana käytetään tiedoston sanojen lukumäärää. Arvioi tämän perusteella, minkä kokoisia tiedostoja ohjelmalla voidaan käsitellä, kun tiedetään, että keskimääräinen sanapituus suomen kielessä on 8.5 merkkiä ja englannissa 5.1 merkkiä. Mittaa myös ohjelmasi suoritusaikoja testisyötteillä ja tee tämän perusteella vastaava arvio syötetiedoston koon ylärajasta.

Ohjelmakoodin lisäksi palautetaan työselostus, jossa kuvataan

1. ratkaisu,
2. käytetyt tietorakenteet ja
3. ohjelman suorituskyvyn analyysi, kuten edellä mainittiin.

Analyysi sisältää siis käytetyn algoritmin analyysin ja ohjelman suoritusaikojen mittaukset. Ohjelmakoodin tulee olla asiallisesti kommentoitu, mutta muuta dokumentointia ei vaadita. Työselostukseen liitetään luonnollisesti myös nimi ja opiskelijanumero. Mikäli haluat, voit antaa työselostuksessa myös palautetta työstä, esimerkiksi tehtävään käytetty työmäärä, työn helpot ja hankalat asiat ja mitä opit työtä laatiessasi. Muutakin palautetta voi antaa.

Muista pakata tiedostot yhdeksi pakkaukseksi ennen palauttamista.

Tehtävän ratkaisu on palautettava viimeistään 15.2.2018.

811312A Data Structures and Algorithms 2017-2018, C assignment

In this assignment, you shall find the 100 most frequently occurring words from a large text file. The program shall be implemented in C language. A continuous string of characters a..z and A..Z, with possible apostrophes ’, is considered a word. You can also include characters å,ä, ö, Å,Ä, and Ö if you wish, but this is not obligatory. Words with uppercase and lowercase letters are considered equal. For example, in the text

Herman Melville’s book Moby Dick starts, as we all know, with the sentence ”Call me Ishmael”.

the words are

herman, melville’s, book, moby, dick, starts, as, we, all, know, with, the, sentence, call, me, **and** ishmael

The name of the text file can be either given as a command-line argument or as an input from the user. The program prints the 100 most frequently occurring words and their frequencies in the file. The words are printed in descending order according to their frequencies.

Because the file can be very large, you need a suitable data structure to store words and their frequencies. This can be, for instance, a hash table or a binary search tree. You do not need to sort all the words according to their frequencies. Hence, you can use maximum heap, for example, to get the 100 most frequent words. Sorting with some fast algorithm is also possible.

Estimate the time complexity of your program, when the size of the input is the number of words in the text file. Based on this, estimate the maximum size of files that can be reasonably processed with your program, when we know that the average word length in English is 5.1 letters. Also, measure the running times of your program with test inputs and make a corresponding estimate for the file size, based on the results of your measurements.

In addition to program code, you shall return a report that describes

1. the solution,
2. the chosen data structures, and
3. analysis of program’s performance as mentioned above.

Analysis contains thus the analysis of the used algorithm and the measurements of program’s running times. The code shall be commented but no other documentation is required. The report shall naturally contain student’s name and student number. If you wish, you can also give feedback in your report.

Remember to zip the files before returning.

The assignment shall be returned by 15.2.2018.