

## Määrittelydokumentti

Pahoittelen, en kerkeä tekemään kunnollista palautusta viikolle 2, koska aloitin koko kurssin vasta periaatteessa tänään. Yritän juosta kiinni mahdollisimman nopeasti. Koodia on turha yrittää kääntää. Vasta hahmottelen algoritmia.

### Mitä ongelmaa ratkaiset ja miksi valitsit kyseiset algoritmit/tietorakenteet

Toteutan Lempel-Ziv [2] algoritmin, jota käytetään tiedostojen tiivistykseen. Valitsin algoritmin, koska se on selkeä ja toimii pohjana monelle muulle edistyneemmälle tiedonpakkausalgoritmille [1]. Projektiani varten teen pakkaus- ja purkutyökalun. Tarkoituksena olisi saada aikaan bufferi, johon tiedostoja luetaan vähitellen, josta dataa voidaan hankkia ikkunaan, eikä koko tiedostoa lueta muistiin. Toinen juttu on, että haluan pystyä määrittelemään bittien lukumäärän “merkissä”, joten en käytä tavua pienempinä yksikkönä.

### Mitä syötteitä ohjelma saa ja miten näitä käytetään

Tiedosto

### Mitä algoritmeja ja tietorakenteita toteutat työssäsi

Toteutan dynaamisen listan ja ehkä jonkinlainen iteraattori sille helpottaisi asiaa.

### Tavoitteena olevat aika- ja tilavaativuudet (m.m. O-analyysit)

Aika vaativuus on kai parhaimmillaan  $O(n \cdot \log(n))$ : käydään kaikki tiedoston bitit läpi sen jälkeen käydään “ikkunan” kokoinen alue läpi kursorin ympäriltä. Tilavaativuus on bufferin koko.

### Lähteet

## References

- [1] David Salomon. *Data Compression: The Complete Reference*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [2] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Information Theory*, 23:337–343, 1977.