1 Algoritmien suorituskyvyt ja aikavaatimusluokat

Harjoitustyössä käytetään Cooley–Tukey FFT - algoritmia. IFFT on toteutettu käyttäen apuna FFT-algoritmia. Vektorit, joiden pituus ei ole 2:n potenssi, käsitellään täyttämällä vectori nollilla seuraavaan toimivaan pituuteen.

Toimintaidea perustuu diskreetin Fourier-muunnoksen ominaisuuteen, jonka mukaan muunnos voidaan esittää summana parillisten- ja parittomien alkioden muodostamista vektoreista.

Diskreetti Fourier-muunnos

$$X_k = \sum_{n=0}^{N-1} x_n e^{(-2\pi ikn/N)} \tag{1}$$

voidaan esittää summina

$$X_k = E_k + e^{(-2\pi ik/N)} O_k$$
$$X_{k+N/2} = E_k - e^{(-2\pi ik/N)} O_k$$

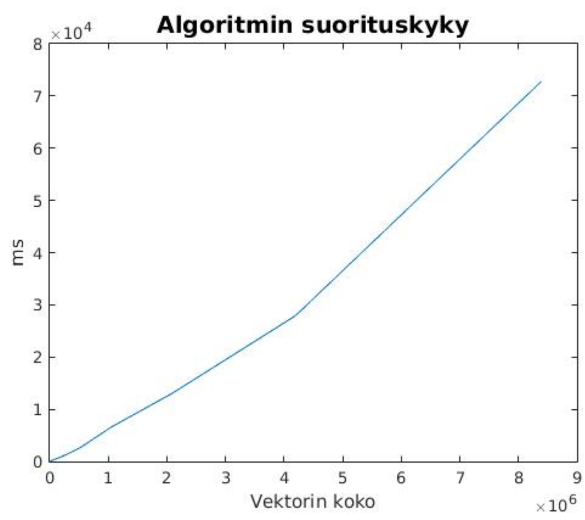
Missä E_k on parillisten alkioden muodostama vektori ja O_k parittomien alkioiden muodostama vektori. Nyt hajoittamalla vektori kahteen osaan ja kutsumalla algoritmia rekursiivisesti saadaan algoritmi toimimaan aikavaatimus luokassa $O(NlogN)$, kun alkuperäisen DFT-algoritmin aikavaatimus on luokkaa $O(N^2)$.

FFTpower2(A)

$N = A.length$
if $N = 1$ **then**
    $A \leftarrow A$
else
    $Aeven \leftarrow FFT(Aeven)$
    $Aodd \leftarrow FFT(Aodd)$
    for $k = 0$ to $k = N/2 - 1$ **do**
        $t \leftarrow Aeven_k$
        $A_k \leftarrow t + \exp(-2\pi ik/N)Aodd_k$
        $A_{k+N/2} \leftarrow t - \exp(-2\pi ik/N)Aodd_k$
    end for
end if

2 Java-toteutuksen suorituskyvyn testausta

Algortimia testattiin eri kokoisilla syötteille. Kuvasta 1 voidaan havaita että algoritmin kuluttama aika kasvaa lähellä lineaarista kasvua. Joten voimme todetta, että aikavaatimus on luokkaa $O(NlogN)$.



Kuva 1: FFT-algoritmi