# O(log n) : Binary search

lähde: <https://en.wikipedia.org/wiki/Binary_search_algorithm>

Binary search algoritmi löytää lajitellusta avainlistasta tai taulukosta. Algoritmi vertaa taulukon keskikohtaa etsittyyn arvoon ja poissulkee listan puolikkaan, joka ei sisällä avainta (suuremman puolikkaan mikäli keskimmäinen arvo on suurempi kuin etsitty arvo ja päin vastoin). Iterointia jatketaan, kunnes taulukkoa ei voi enää puolittaa tai kunnes arvo on löytynyt.

Koska syötemäärä puolitetaan jokaisella iteraatiolla, ei algoritmi voi pahimmassakaan tapauksessa (mikäli arvoa ei löydy tai se sijaitsee lehtisolmussa) ylittää log2n -iteraatio, joten se kuuluu kertaluokkaan O(log n)

# O(n logn): Merge sort

lähde: <https://en.wikipedia.org/wiki/Merge_sort>

Merge sort algoritmi on ns. “divide and conquer” algoritmi, eli se jakaa isomman onlgelman joukoksi pienempiä osaongelmia, jotka se sitten ratkaisee yksitellen. Merge sort perustuu tietoon, että kaksi lajiteltua taulukkoa on helpompi lajitella kuin yksi lajittelematon, koska silloin jo yhden vertauksen perusteella solun paikka tiedetään.

Jokaisessa iteraatiossa täytyy tehdä korkeintaan x\*2 verran vertailuja (x = osataulukon pituus).

# O(n2): Selection sort

”Selection sort”-lajittelualgoritmi on kertaluokan O(n2) algoritmi. Algoritmi on ns. paikallaan toimiva algoritmi eli se ei vaadi ylimääräistä tilaa. Se toimii siirtämällä

Se käy taulukon läpi alkio kerrallaan ja vertaa kyseistä alkiota aiempiin alkoihin, mikäli uusi alkio on aikaisempia pienempi, vaihdetaan niiden paikkaa. Koska aikaisemmat alkiot ovat lajiteltu, voidaan siirtyä uuteen iteraatioon heti suuremman alkion löydyttyä.

lähde: <https://en.wikipedia.org/wiki/Selection_sort>

Selection sort algoritmia on helppo analysoida, koska sen vertailujen määrä ei riipu aineiston järjestyksestä. Vertailujen määrän voi laskea yhtälöllä: ½(n2-n).

Syötteen kasvaessa todella suureksi ainoastaan eksponentilla on merkittävä vaikutus laskentaresursseihin, joten n2-termi määrää kertaluokan, eli kuuluu algoritmi kertaluokkaan O(n2).

# O(vakion): Salasanan murtaminen ”brute force” menetelmällä