Bubble Sort

**Egenskaper:**

Komplexitet i värsta/average fall: O(n2)

Komplexitet i bästa fall: O(n)

**Summering:**

Iterera genom arrayen och byt plats på varje par som ligger i oordning. Upprepa tills inga par ligger i ordning. Namnet kommer från att det högsta elementet “bubblar upp” till slutet av arrayen.

Insertion Sort

**Egenskaper:**

Komplexitet i värsta/average fall: O(n2)

Komplexitet i bästa fall: O(n)

Likt hur en människa skulle sortera en pokerhand.

**Summering:**

Ha en sorterad lista (V) och en osorterad lista (H) och flytta kontinuerligt element från den osorterade listan H till den sorterade V genom att sätta in dem på rätt ställe i V.

Selection Sort

**Egenskaper:**

Komplexitet i värsta/average fall: O(n2)

Komplexitet i bästa fall: O(n2)

Logisk och lätt att förstå men ineffektiv i praktiken.

**Summering:**

Leta upp det minsta elementet i den osorterade arrayen och byt plats med det första och markera det första elementet som sorterat. Upprepa proceduren med resten av arrayen.

Merge Sort

**Egenskaper:**

Komplexitet i värsta/average/bästa fall: O(n log n)

Rekursiv, divide and conquer. En av de mest effektiva sorteringsalgoritmerna! Kräver ofta extra O(n) minne.

**Summering:**

Dela upp listan på häften, upprepa tills varje lista är ett element. Sammanför (merge) sedan varje par av sorterad lista till en.

QuickSort

**Egenskaper:**

Komplexitet i average/bästa fall: O(n log n)

Komplexitet i värsta fall: O(n2)

Rekursiv, divide and conquer. En av de mest effektiva sorteringsalgoritmerna! Dåligt värsta-fall!

**Summering:**

Välj ett pivotelement och partitionera arrayen i två delar där alla element till vänster är mindre eller lika med pivotelementet och till höger är alla element större än pivotelementet och pivotelementet i mitten. Upprepa rekursivt för båda sidorna av arrayen.