

## Detaljert eksempel av Insertion Sort

Skal sorteres (reverse order):

```
a[]      [ '-1', 'Y', 'S', 'E', 'A' ]
Indeks:  [  0,   1,   2,   3,   4 ]
Verdi:   [  0,  25,  19,   5,   1 ]
N = 4
```

```
// Outer loop 1.
for(i = 2; i <= N; i++) {
    v = a[i];           // v = a[2] => v = 'S'
    j = i;              // j = 2

    // Inner loop 1.
    while (a[j-1] > v) { // a[1] > v => 'Y' > 'S' (25 > 19)? Ja
        a[j] = a[j-1];  // a[2] = a[1] => a[2] = 'Y'
        j--;            // j = 1
    }
    // Inner loop 2 sjekkes, men a[0] = -1 (sentinal key) og -1 > 'S' stemmer ikke.
    // Inner loop er ferdig.
    a[j] = v;           // a[1] = 'S'
}
```

```
// Etter Inner loop 1 og Outer loop 1 har kjørt, har vi følgende:
// a[2] = 'Y' og a[1] = 'S'
// Arrayen er nå: [ '-1', 'S', 'Y', 'E', 'A' ]
```

```
// Outer loop 2. (i var 2)
for (i = 2; i <= N; i++) { // i ble inkrementert til 3 i forrige loop

    v = a[i];           // v = a[3] => v = 'E'
    j = i;              // j = 3

    // Inner loop 1.
    while (a[j-1] > v) { // a[2] > v => 'Y' > 'E' (25 > 5)? Ja
        a[j] = a[j-1];  // a[3] = a[2] => a[3] = 'Y'
        j--;            // j = 2
    }
    // Inner loop 2 sjekkes. a[j-1] > v => 'S' > 'E' (19 > 5)? Ja
    // Inner loop 2 kjører
    a[j] = a[j-1];      // a[2] = a[1] => a[2] = 'S'
    j--;                // j = 1.
    // Inner loop 3 sjekkes, men a[0] er sentinal key, loop stopper
    a[j] = v;           // a[1] = 'E'
}
```

```
// Etter inner loop 1, 2 og Outer loop 2 har kjørt, har vi følgende:
// a[3] = 'Y', a[2] = 'S' og a[1] = 'E'
// Arrayen er nå: [ '-1', 'E', 'S', 'Y', 'A' ]
```

```
// Outer loop 3. (i var 3)
for (i = 2; i <= N; i++) { // i ble inkrementert til 4 i forrige loop
    v = a[i];           // v = a[4] => v = 'A'
    j = i;              // j = 4

    // Inner loop 1.
    while (a[j-1] > v) { // a[3] > v => 'Y' > 'A' (25 > 1)? Ja
        a[j] = a[j-1];  // a[4] = a[3] => a[4] = 'Y'
        j--;            // j = 3
    }
    // Inner loop 2 sjekkes. a[j-1] > v => 'S' > 'A' (19 > 1)? Ja
```

```

// Inner loop 2 kjører
  a[j] = a[j-1];           // a[3] = a[2] => a[3] = 'S'
  j--;                     // j = 2
// Inner loop 3 sjekkes. a[j-1] > v => 'E' > 'A' (5 > 1)? Ja
// Inner loop 3 kjører
  a[j] = a[j-1];           // a[2] = a[1] => a[2] = 'E'
  j--;                     // j = 1
// Inner loop 4 sjekkes, men a[0] er sentinal key, loop stopper
  a[j] = v;                 //a[1] = 'A'
}

// Etter at Inner loop 1, 2, 3 og Outer loop 3 har kjørt, har vi følgende:
// a[4] = 'Y', a[3] = 'S', a[2] = 'E' og a[1] = 'A'
// Arrayen er nå: [ '-1', 'A', 'E', 'S', 'Y' ]

// N var satt til 4, vi har loopet 3 ganger, og i har inkrementert seg til 5
// 5 <= 4? Nei og Outer loop 4 vil ikke kjøre
// Vi er ferdig med sorteringen

// Vi brukte 9 comparisons og 6 swaps

```