```
let C[0..n − 1] be a new array;
for i = 0 to n − 1 do
    C[i] ← 0;
end
for i = 0 to n − 2 do
    for j = i + 1 to n − 1 do
        if A[i] < A[j] then
            C[j] ← C[j] + 1;
        end
        else
            C[i] ← C[i] + 1;
        end
    end
end
for i = 0 to n − 1 do
    B[C[i]] ← A[i];
end
return B;
```

Algorithm 14: comparison-counting-sort($A[0..n-1]$)

$$T(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (n-1-(i+1)+1) = \sum_{i=0}^{n-2} (n-i-1) = \sum_{i=1}^{n-1} i = \frac{(n-1).n}{2} = \Theta(n^2)$$

$K = h - l$

```
let C[0..h − l] be a new array;
for i = 0 to h − l do                      } Θ(K)
    C[i] ← 0;
end
for i = 0 to n − 1 do                      } Θ(n)
    C[A[i] − l] ← C[A[i] − l] + 1;
end
for j = 1 to h − l do                      } Θ(K)
    C[j] ← C[j] + C[j − 1];
end
for i = n − 1 downto 0 do                  } Θ(n)
    j ← A[i] − l;
    B[C[j] − 1] ← A[i];
    C[j] ← C[j] − 1;
end
return B;
```

Algorithm 15: counting-sort($A[0..n-1], l, h$)

$\Rightarrow T(n) = \Theta(n + K)$

Se $K = O(n)$ então

$T(n) = \Theta(n)$