**Exercise 1:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | log2(n) | s(n) | 5 | 2n | 1/n | n | en |
| log2(n) | Θ | - | *O* | ω | *O* | ω | ω |
| s(n) | - | Θ | - | - | - | - | - |
| 5 | Ω | - | Θ | Ω | Ω | Ω | Ω |
| 2n | *o* | - | *O* | Θ | *o* | *o* | Θ |
| 1/n | Ω | - | *O* | ω | Θ | ω | ω |
| n | *o* | - | *O* | ω | *o* | Θ | ω |
| en | *o* | - | *O* | Θ | *o* | *o* | Θ |

**Exercise 2:**

Let A be an array with elements A[1];…;A[n]. Consider the following algorithm:

**for** i = 1;…; n - 1 **do**

Find the minimum of the elements A[i];…;A[n] and the corresponding index j.

**if** i 6= j **then**

Swap A[i] and A[j].

**end if**

**end for**

**return** A

a) The algorithm solves the problem of “Sorting”.

Invariant:

A[j] ≤ A[i]

b) The algorithms carries out (n-1)2 comparisons.

c) The array elements are swapped (Minimum: 0, Maximum: n-1)

Minimum swapping array: [1, 2, 3, 4, 5, 6]

Maximum swapping array: [6, 5, 4, 3, 2, 1]