E2011: Theoretical fundamentals of computer science
Introduction to algorithms - Additional exercises

Vlad Popovici, Ph.D.

Fac. of Science - RECETOX

Problem 1

Search problem

Given a sequence of n numbers, $A = [a_1, \ldots, a_n]$ and a value v, find

1. whether v appears in A and, if yes, output its position, otherwise output "value not found" message;
2. whether v appears in A and, if yes, output its position, otherwise output the closest value in A to v

- identify the input and output
- express the solution

Algorithm 1 Find value in a sequence - part 1

Input: $n \in \mathbb{N}$, $A = [a_1, \ldots, a_n], v \in \mathbb{R}$
Output: i such that $a_i = v$ or text

```
for i = 1, ..., n do
    if a_k = v then
        return i
    end if
end for
print "value not found!"
```

Problem 2

Selection sort

Implement the following sequence sorting algorithm for n values $A = [a_1, \ldots, a_n]$: first find the smallest element of A and exchange it with the element in a_1. Then find the second smallest element of A, and exchange it with a_2. Continue in this manner for the first $n - 1$ elements of A.
What needs to be changed to obtain a decreasing ordered sequence?

Solution to Problem 2

Algorithm 2 Find value in a sequence - part 1

Input: $n \in \mathbb{N}$, $A = [a_1, \ldots, a_n] \in \mathbb{R}$
Output: ordered sequence A

```
for i = 1, ..., n − 1 do
    min ← i
    for j = i + 1, ..., n do
        if a_j < a_min then
            min ← j
        end if
    end for
    if min ≠ i then
            ▷ swapping values is needed only if a_i is not already minimum
        tmp ← a_i                    ▷ these 3 lines are for swapping values
        a_i ← a_min
        a_min ← tmp
    end if
end for
```

Problem 3

Binary addition

Consider two numbers A and B represented in binary as two vectors of bits $A = [a_1 a_2 \dots a_n]$ and $B = [b_1 b_2 \dots b_n]$ with most significant bit being at position 1 and least significant one at position n. Write the pseudocode to perform the addition of the two numbers, such that the result $C = A + B$ is represented as a $n + 1$ vector of bits $C = [c_1 c_2 \dots c_{n+1}]$.

Solution to Problem 3

Input: $n \in \mathbb{N}$, $A = [a_1 a_2 \ldots a_n]$, $B = [b_1 b_2 \ldots b_n]$
Output: $C = A + B$, $C = [c_1 c_2 \ldots c_{n+1}]$

```
carry ← 0
for i = n, n − 1, . . . , 1 do
    c_{i+1} ← (a_i + b_i + carry) mod 2
    if a_i + b_i + carry ≥ 2 then
        carry ← 1
    else
        carry ← 0
    end if
end for
c_1 ← carry
```