INFO2050 - Advanced computer programming

Exercise session 1: Pseudo-code and complexity

Jean-Michel BEGON

September 2014

Exercice 1

What is this function doing?

```
MYSTERY(A)
if A.length < 2
    return True
else
    if A[1] == A[A.length]
        return MYSTERY(A[2..A.length − 1])
    else
        return False
```

Exercise 2

(a) Write the pseudo-code of an iterative function which determines the minimum value of an array. Write the same function in a recursive way.

(b) Write the pseudo-code of a recursive function which computes the following recurrence:

$$\begin{aligned} T(i,j) &= T(i-1,j) + T(i,j-1) \\ T(i,1) &= 1 \quad \forall i > 0 \\ T(1,j) &= 1 \quad \forall j > 0 \end{aligned}$$

Exercise 3

(a) Algorithm A requires $10n^3$ operations to solve a problem. Algorithm B solves it in $1000n^2$ operations. What is the fastest algorithm.

(b) Algorithm A requires $32n \log_2 n$ operations to solve a problem. Algorithm B solves it in $3n^2$ operations. What is the fastest algorithm.

Exercise 4

Let an algorithm whose execution time for $N = 1000$, 2000, 3000 and 4000 be $5s$, $20s$, $45s$ et $80s$ respectively. Give an estimation of the required time for $N = 5000$.

Exercise 5

(a) Show that $2n + 100$ is $\Theta(n)$.

(b) Show that $5n^2 + 500n + 5000$ is $\Theta(n^2)$.

(c) Show that 2^{n+1} is $\Theta(2^n)$.

(d) Explain why the sentence *"The execution time of algorithm A is at least $O(n^2)$"* does not make sense.

(e) Show that the execution time of an algorithm is $\Theta(g(n))$ if and only if the execution time is both $O(g(n))$ and $\Omega(g(n))$.

(f) Give an example of a function $f(n)$ which is neither $O(n)$ nor $\Omega(n)$.

Exercise 6

Sort these function by increasing order of complexity (regarding the $\Theta(.)$, $O(.)$ and $\Omega(.)$ operators).

$n \log_2 n$	$\frac{4}{n}$	$\sqrt{n}$	2^{2^n}
$\log_2 \log_2 n$	$8n^3$	$8^{\ln n}$	$\frac{n}{2+n}$
$\log_2 n^7$	$5^{\ln \log_2 n}$	$(\log_2 n)^3$	$\frac{n}{\log_2(2+n)}$

Exercise 7

For each of the following pseudo-codes, determine what is the algorithm doing and what is the asymptotic complexity in terms of n. (Be precise in the notations).

```
CODE1(n)
limit = n * n
sum = 0
for i = 1 to limit
        sum = sum + 1
return sum
```

```
CODE2(n)
i = 1
limit = n * n * n
sum = 0
while i < limit
        sum = sum + 1
        i = i * 2
return sum
```

```
Code3(a, b, c, n)
for i = 1 to n
    for j = 1 to n
        a[i][j] = 0
        for k = 1 to n
            a[i][j] = a[i][j] + b[i][k] * c[k][j]
```

Exercise 8

Let A be an array of n values sorted in ascending order. Our purpose is to determine if a value b is present in A.

(a) Write a pseudo-code of a brute-force algorithm for finding b. What is its complexity in the best/average/worst case ?

(b) Give an dichotomic algorithm. What is its best/average/worst case complexity ?

Exercise 9

Let A be an array of N integers where each integer in the $1..N$ interval appears exactly once except for an integer appearing twice and one missing. Give an linear-time algorithm for finding the missing integer which takes a memory space of at most $O(1)$.