AI - Assignment 2

Patrick Nagel: patrick.nagel@h-brs.de
Amirhossein Pakdaman: amirhossein.pakdaman@smail.inf.h-brs.de

Due Date: 20.10.2019

You can get help from the pdf file: *cormen_introalgorithms* provided in LEA to solve the following assignments.

1. Why is a special notation needed to classify algorithms? Doesn't it suffice to merely measure the runtime in seconds? Explain.

2. Let $g : N \rightarrow R+$ be an arbitrary function. The set of functions $f : N \rightarrow R+$, which do not grow faster than the function g after a specific point n_0 , is denoted as $O(g(n))$. More specifically: $O(g(n)) := f(n) | \exists\, c \,\epsilon\, R \,and\, \exists\, n_0 \,\epsilon\, N : 0 \leq f(n) \leq cg(n)\ \forall\, n \geq n_0$

   Prove the following:

   - $f(n) = 100n^2 \in O(n^2)$
   - $f(n) = n^6 + 100n^5 \in O(n^6)$

3. What is the running time of the following python-code in O-Notation? Assume, that *ANY_CONST* is an arbitrary constant in your program, which receives a 2d array *arr* as parameter.

```
sum = 0
for i in range(0, J):
    for j in range(0, K):
        if arr[i][j] <= ANY_CONST:
            sum = sum + arr[i][j]
print(sum)
```

4. For each function $f(n)$ and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes $f(n)$ microseconds. Please briefly mention how you do the calculations for items: $log(n)$, 2^n, and $n\ log(n)$.

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
$\sqrt{n}$							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							