Lab01-Algorithm Analysis

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2021.

$*$ If there is any problem, please contact TA Haolin Zhou. Also please use English in homework.

$*$ Name:Yanjie Ze   Student ID: 519021910706   Email: zeyanjie@sjtu.edu.cn

1. *Complexity Analysis.* Please analyze the time and space complexity of Alg. 1 and Alg. 2.

Algorithm 1: QuickSort

```
Input: An array $A[1, \cdots, n]$
Output: $A[1, \cdots, n]$ sorted nondecreasingly
$pivot \leftarrow A[n]$; $i \leftarrow 1$;
for $j \leftarrow 1$ to $n - 1$ do
    if $A[j] < pivot$ then
        swap $A[i]$ and $A[j]$;
        $i \leftarrow i + 1$;
swap $A[i]$ and $A[n]$;
if $i > 1$ then
    QuickSort($A[1, \cdots, i - 1]$);
if $i < n$ then
    QuickSort($A[i + 1, \cdots, n]$);
```

Algorithm 2: CocktailSort

```
Input: An array $A[1, \cdots, n]$
Output: $A[1, \cdots, n]$ sorted nonincreasingly
$i \leftarrow 1$; $j \leftarrow n$; $sorted \leftarrow false$;
while not $sorted$ do
    $sorted \leftarrow true$;
    for $k \leftarrow i$ to $j - 1$ do
        if $A[k] < A[k + 1]$ then
            swap $A[k]$ and $A[k + 1]$;
            $sorted \leftarrow false$;
    $j \leftarrow j - 1$;
    for $k \leftarrow j$ downto $i + 1$ do
        if $A[k - 1] < A[k]$ then
            swap $A[k - 1]$ and $A[k]$;
            $sorted \leftarrow false$;
    $i \leftarrow i + 1$;
```

(a) Fill in the blanks and **explain** your answers. You need to answer when the best case and the worst case happen.

Algorithm	Time Complexity[1]	Space Complexity
QuickSort		
CocktailSort		

[1] The response order can be given in *best*, *average*, and *worst*.

(b) For Alg. 1, how to modify the algorithm to achieve the same expected performance as the **average** case when the **worst** case happens?

2. *Growth Analysis.* Rank the following functions by order of growth with brief explanations: that is, find an arrangement $g_1, g_2, \ldots, g_{15}$ of the functions $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots, g_{14} = \Omega(g_{15})$. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$. Use symbols "=" and "$\prec$" to order these functions appropriately. Here $\log n$ stands for $\ln n$.

1	n	$\log n$	$\log(\log n)$	$n \log n$
$\log_4 n$	2^n	4^n	$2^{\log n}$	2^{2^n}
$\log(n!)$	$n!$	$(2n)!$	$n^{1/2}$	n^2

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.