CS222 Homework 1

Algorithm Analysis & Deadline: 2020-09-21 Monday 24:00

Exercises for Algorithm Design and Analysis by Li Jiang, 2020 Autumn Semester

1. Prove that $\log(\log n) = o(n^k)$, where k is a positive constant. (ps: $\log n$ refers to $\log_2 n$.)

2. Prove that for any integer $n^2 - 1 > 3$, there is a prime p satisfying $n! > p > n$.

3. Assume that there is a recurrence formula as follows:

$$D(x) = \begin{cases} 1, & if\ x == 1 \\ 3D(x/4) + x - 2, & if\ x \geq 2 \end{cases}$$

   Please deduce the non-recursive expression of $D(x)$ and point out its asymptotic complexity.

4. Use the minimal counterexample principle to prove that for any integer $n > 10$, there exist integers $i_n \geq 0$ and $j_n \geq 0$, such that $n = i_n \times 3 + j_n \times 4$.

5. Analyze the **average** time complexity of QuickSort in Alg. **??**.

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 $A[j]$;
        $i \leftarrow i + 1$;
    end
end
swap $A[i]$ and $A[n]$;
if $i > 1$ **then** QuickSort($A[1, \cdots, i - 1]$);
if $i < n$ **then** QuickSort($A[i + 1, \cdots, n]$);

6. Rank the following functions by order of growth with explanations: that is, find an arrangement $g_1, g_2, \ldots, g_k$ of the functions $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots, g_{k-1} = \Omega(g_k)$. 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. (ps: $\log n$ refers to $\log_2 n$.)

$2^{\log n}$	$(\log n)^{\ln n}$	n^2	$n!$	$(n-1)!$
2^n	n^3	$\log^2 n$	e^n	2^{2^n}
$\log\log n$	$(n+1)\cdot 2^n$	n	$\log(n^2 - n)$	$2^{\ln n}$

Remark: You need to upload your .pdf file.