Martin Ziegler *Total 16 pts + 6 bonus pts*

Donghyun Lim, Dongseong Seon, Ivan Koswara

CS204

Spring 2018, Homework #5

Problem 1. *5 + 2 pts*

There are three algorithms: algo1, algo2, and algo3. Each of them requires different number of clock cycles to complete a task. Algo1 requires 2^n clock cycles to complete a task of input size n. Algo2 requires n^3 clock cycles to complete a task of input size n. Algo3 requires $n \log_2 n$ clock cycles to complete a task of input size n. The three algorithms run on different CPUs. Algo1 runs on a CPU of clock rate at 1 GHz. Algo2 runs on a CPU of clock rate at 1kHz. Algo3 runs on a CPU of clock rate at 0.1 Hz.

a) Fill out the following table.
b) Which is more important between CPU speed improvement and improvement in asymptotic number of steps required for an algorithm? Give an explanation.

	Running time(sec)		
input size	algo1	algo2	algo3
1			
10			
100			
1000			
10000			

Problem 2. *2 + 2 + 2 pts*

In Chapter 3.1 Example 6, there is Algorithm 6. For a given amount of money, the algorithm uses coins of available types to express that amount of money: the sum of chosen coins should equal the given amount of money. The algorithm tries to minimize the number of coins with a greedy approach: given k units of money, while having available coins for each type $c_1 < \cdots < c_n$, the algorithm chooses one coin c_i satisfying $c_i \leq k < c_{i+1}$. The same process repeats with remaining money $k - c_i$, until the remaining money becomes 0.

a) Suppose we have infinitely many coins for each type: 100 cents, 50 cents, 10 cents, 5 cents, and 1 cent. Using that algorithm, describe how to express 486 cents.
b) Suppose that now we have infinitely many coins for each type: 120 cents, 100 cents, 20 cents, 5 cents, and 1 cent. Using that algorithm, describe how to express 327 cents.
c) Does b) give the minimum number of coins as a solution? If not, give a solution with minimum number of coins.

Problem 3. *3 + 3 + 3 pts*

Prove or disprove each statement.

a) $f(n)$ is $O(g(n))$ if and only if $g(n)$ is $\Omega(f(n))$.
b) (Bonus) If $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$, then $f(n)$ is $O(h(n))$.
c) (Bonus) If $f(n)$ is $O(g(n))$, then $2^{f(n)}$ is $O(2^{g(n)})$.