CSE530: Algorithms & Complexity
Exercise Set 1

Antoine Vigneron

February 28, 2018

Question 1. For each function $f(n)$ below, give a tight bound using the $\Theta(\cdot)$ notation. Your answer should be chosen from the following list: $\Theta(1)$, $\Theta(n)$, $\Theta(\log n)$, $\Theta(n \log n)$, $\Theta(n!)$, $\Theta(n^2)$, $\Theta(2^n)$, $\Theta(n^3)$, $\Theta(\sqrt{n})$, $\Theta(n \log^2 n)$, $\Theta(3^n)$. No justification is needed.

	$f(n)$	Answer
(a)	$10(n+1)(n-5)$	
(b)	$n^2 + 2^n$	
(c)	$n \log(n^2)$	
(d)	$\sqrt{n} + \log n$	
(e)	$10^{23} + \log n$	
(f)	$n \log_9 n + n^2$	
(g)	$(\sqrt{n} \log n)^2$	
(h)	$(n+1)^3 - (n-1)^3$	
(i)	$\dfrac{n+\sqrt{n}}{n+1}$	
(j)	$2n + n \sin(n)$	

2. Let $f(n)$ and $g(n)$ be two functions that are nonnegative for every $n \geqslant n_0$, where n_0 is a constant. Using the basic definition of the Θ-notation, prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

3. Suppose that $f(n) = O(g(n))$, and that there exists a constant n_1 such that $n \geqslant n_1$ implies that $f(n) \geqslant 2$ and $g(n) \geqslant 2$. Prove that $\log f(n) = O(\log g(n))$. (Remember that in this course, log means $\log_2$.)

4. Given an input array $A[1 \ldots n]$ and a key x, the *searching problem* is to decide whether x is stored in A and if so, return i such that $x = A[i]$. The brute force approach for searching, which scans through the whole input array, is called *linear search.*

Write the pseudocode for linear search, and prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties

5. Consider the problem of adding two n-bit binary integers, stored in two n-element arrays A and B. The sum of the two integers should be stored in binary form in an $(n + 1)$-element array C . State the problem formally and write pseudocode for adding the two integers.

6. How can we modify almost any algorithm to have a good best-case running time?