CSc 30400 Introduction to Theory of Computer Science
9th Homework Set

1. Place the following functions in a table in decreasing order. For example, if $f(n) = o(g(n))$ then $g(n)$ will be placed above $f(n)$ in the table. Functions having a Θ relation ($f(n) = \Theta(g(n))$) should be placed in the same level.

   (a) $2n^2 + 5n - 4$

   (b) $2^{10}n$

   (c) $2^{(n+1)} + 6n$

   (d) $n!$

   (e) n

   (f) $\log_2(n!)$

   (g) $\frac{1}{5}n^2$

   (h) 2^n

   (i) $\log_2 n$

   (j) $\log_5 n^2$

2. We said in class that if we do care about space complexity we prefer to use the alphabet $\Sigma = \{0, 1\}$ and to denote numbers in binary. That is because we need just $\log n$ boxes in order to express the number n in binary, where in unary we need n. The task of this exercise is to explore the difference on the running time and the used space in those two cases. The functions we are comparing are the following:

   - $f(n) = 2n$
   - $g(n) = n + 1$

   (a) Assume that the input-output alphabet is $\Sigma = \{1\}$ and the numbers are expressed in unary. For each function design a DTM that computes it (try to be as efficient as possible in designing the machines). Determine the time and space complexity of each machine. You don't have to be exact for the time complexity of the machine that computes f (use the O-notation).

(b) Now assume that the input-output alphabet is $\Sigma = \{0, 1\}$ and the numbers are expressed in binary. Design a DTM that computes $f(n)$. This machine should take n in binary as input and after the computation should output $2n$ in binary again (hint: multiplying a binary number by 2 is just like multiplying a decimal number by 10). Compute the time and space complexity of this machine. Be exact (don't use the O-notation)! Which is more efficient (between this one and the one that uses unary)?

(c) Using the same input-output alphabet $\Sigma = \{0, 1\}$ as above design a DTM for $g(n)$. Compute the time and space complexity in the worst case (which is the worst case?). Be exact (don't use the O-notation)! Which is more efficient (between this one and the one that uses unary)?