CSCI 570 - Spring 2022 - HW2

Due January 26th

1 Graded Problems

1. What is the tight upper bound to the worst-case runtime performance of the procedure below?

   $c = 0$
   $i = n$
   while $i > 1$ **do**
       for $j = 1$ to i **do**
           $c = c + 1$
       end for
       $i = \text{floor}(i/2)$
   end while
   return c

2. Arrange these functions under the O notation using only $=$ (equivalent) or $\subset$ (strict subset of):

   (a) $2^{\log n}$
   (b) 2^{3n}
   (c) $n^{n \log n}$
   (d) $\log n$
   (e) $n \log\left(n^2\right)$
   (f) n^{n^2}
   (g) $\log(\log(n^n))$

   E.g. for the function n, $n+1$, n^2, the answer should be

   $$O(n+1) = O(n) \subset O(n^2).$$

3. Given functions f_1, f_2, g_1, g_2 such that $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$. For each of the following statements, decide whether you think it is true or false and give a proof or counterexample.

   (a) $f_1(n) \cdot f_2(n) = O\left(g_1(n) \cdot g_2(n)\right)$

   (b) $f_1(n) + f_2(n) = O\left(\max\left(g_1(n), g_2(n)\right)\right)$

   (c) $f_1(n)^2 = O\left(g_1(n)^2\right)$

   (d) $\log_2 f_1(n) = O\left(\log_2 g_1(n)\right)$

4. Given an undirected graph G with n nodes and m edges, design an $O(m+n)$ algorithm to detect whether G contains a cycle. Your algorithm should output a cycle if G contains one.

2 Practice Problems

1. Solve Kleinberg and Tardos, **Chapter 2, Exercise 6.**

2. Solve Kleinberg and Tardos, **Chapter 3, Exercise 6.**