CS200 - Problem Set 6

1. Prove Algorithm 1 correctly outputs a list of the prime factors of an integer. The prime factors of n are a list of primes whose product is n. For example for input 60, the algorithm outputs: “2,2,3,5”, since $2, 3, 5$ are all prime, and $2 \times 2 \times 3 \times 5 = 60$. (To prove this result, you need to prove that the outputs are all prime, and that the product of all of the outputs gives you the input.)

Input : An integer n such that $n \geq 2$
Output: String of the prime factors of n

```
/* Recursive Step                                  */
d = 2;
while ¬d|n do
|   d = d + 1;
end
if d = n then
|   return “n”;
else
|   return “d,”+Factor(n/d). // + concatenates strings
end
```

Algorithm 1: `Factor`(n)

2. Max-Weight-Independent-Set is a problem with many applications, but is used in particular to schedule communications among cell towers. You might study it in CS 302. The following is a part of the correctness proof of the algorithm. Let $[m] = \{1, 2, 3, \ldots, m\}$. An independent set S on $[m]$ satisfies: $S \subseteq [m]$ and if $i \in S$, then $i + 1 \notin S$ and $i - 1 \notin S$. Given a function $f : [n] \to \mathbb{N}$ and $1 \leq m \leq n$, we would like to find the independent set S on $[m]$ such that $\sum_{i \in S} f(i)$ (the weight) is as large as possible. $\sum_{i \in S} f(i)$ is the sum of the values of f of all of the elements in the set S. Let S_m^* be the independent set with maximum weight on $[m]$.

   For example, if $f(1) = 4$, $f(2) = 3$, $f(3) = 2$, and $f(4) = 5$, then $S_4^* = \{1, 4\}$, and $S_3^* = \{1, 3\}$.

   (a) Prove that if there is a unique max-weight independent set S_m^* and if $m \notin S_m^*$, then $S_m^* = S_{m-1}^*$.

   (b) Prove that if there is a unique max-weight independent set S_m^* and if $m \in S_m^*$, then $S_m^* = \{m\} \cup S_{m-2}^*$.

3. How many bit strings of length 8 contain at least 6 consecutive 0’s?

4. What is the tightest (smallest) big-O bound that you can prove on e^{-x}: $O(1)$, $O(x)$, or $O(2^x)$. Give a k and C for your response.

5. (a) What is the largest number of edges possible in an undirected graph with n vertices and no self-loops. (A self loop is an edge from a vertex back to itself). Please use a counting rule to count possible edges.
   (b) What is the largest number of edges possible in an undirected graph with n vertices and self-loops.
   (c) What is the largest number of edges possible in a directed graph (where every edge is of the form (a, b) if a and b are vertices) with n vertices and no self-loops.
   (d) What is the largest number of edges possible in a directed graph with n vertices and self-loops.

6. How long did you spend on this homework?