CS200 - Problem Set 6

Due: Tuesday, April. 3 to Canvas before 10:10 am

1. The floor and ceiling functions come up often in computer science. Their domain is the real numbers and their codomain is the integers. $\lfloor x \rfloor$ ("the floor of x") is the largest integer less than or equal to x. $\lceil x \rceil$ ("the ceiling of x") is the smallest integer greater than or equal to x. One reason floor and ceiling appear often in computer science is because computers are much better at dealing with integers than real numbers (why is that?), and so often deal with real numbers by turning them into integers.

   (a) [**2 points**] What is $\lfloor -\sqrt{2} \rfloor$?

   (b) [**3 points**] Is the ceiling function surjective? Explain why.

   (c) [**3 points**] Is the floor function injective? Explain why.

   (d) [**11 points**] Prove true or prove false: $\forall x \in \mathbb{R}, \lceil (\lfloor x \rfloor) \rceil = \lfloor x \rfloor$.

   (e) [**11 points**] Prove true or prove false: $\forall x \in \mathbb{R}, \lfloor 2x \rfloor = 2\lfloor x \rfloor$.

2. Suppose a procedure involves m tasks, where task i can be completed in n_i ways. (So task 1 can be completed in n_1 ways, task 2 can be completed in n_2 ways, etc.) Prove (think about the proof technique you might use) using the product rule that there are

$$\prod_{i=1}^{m} n_i \tag{1}$$

   ways of completing the procedure. Note that if $(a_j, a_{j+1}, a_{j+2}, \ldots, a_k)$ is an ordered set of real numbers, then:

$$\prod_{i=k}^{j} a_i = a_k \times a_{k+1} \times \cdots \times a_{j-1} \times a_j. \tag{2}$$

3. (3 points) How many surjective functions are there from set A to set B if $|A| = n$ and $|B| = 2$? Recall that a function $f : A \to B$ is surjective iff $\forall b \in B\, \exists a \in A,\ f(a) = b$

4. On your worksheet, I introduced relations. A special type of relation is called an equivalence relation. An equivalence relation is a relation from a set A to A ($R \subset A \times A$) and has the following properties:

   - **Reflexive Property:** $\forall a \in A,\ (a, a) \in R$
   - **Symmetric Property:** $\forall a, b \in A,\ (a, b) \in R \leftrightarrow (b, a) \in R$
   - **Transitive Property:** $\forall a, b, c \in A,\ (a, b) \in R \land (b, c) \in R \to (a, c) \in R$

These relations are called *equivalence* relations because these three properties are also properties of "equals:" $a = a$ for all a, if $a = b$, then $b = a$, and if $a = b$ and $b = c$, then $a = c$.

For each of the following relations, either prove it is an equivalence relation, or prove it is not an equivalence relation.

(a) **[11 points]** Let S be the set of strings. Then for $s, t \in S$, $(s, t) \in R$ if and only if $s = t$, or s and t each have at least n characters, and the first n characters of the two strings are the same. (This relation is important for programming languages. For example, in the C programming language, the compiler only looks at the first 31 characters of a variable name.)

(b) **[11 points]** Let $R \subset \mathbb{R} \times \mathbb{R}$. Then for $a, b \in \mathbb{R}$, we have $(a, b) \in R$ if and only if $|a - b| \leq 1$.

5. **[11 points]** 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 $2 \times 2 \times 3 \times 5 = 60$. Hint: while proving the inductive step, you should have two cases for the if/else statement.

```
Input  : An integer n such that n ≥ 2
Output: String of the prime factors of n
/* Recursive Step                                    */
d = 2;
while n%d ≠ 0 do
    d+ = 1;
end
if d == n then
    return "n";
else
    return Factor(d)+Factor(n/d). // "+" concatenates
        strings
end
```

Algorithm 1: `Factor`(n)

6. How long did you spend on this homework?