

CS200 - Problem Set 7

In this problem set, I ask you to prove statements without telling you how to prove them. Make sure you spend some time thinking (or experimenting) with possible approaches. If your first approach doesn't seem to be working, try a different proof technique.

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, and so often deal with real numbers by turning them into integers.
 - (a) **[6 points]** Why are computers better at dealing with integers instead of real numbers?
 - (b) **[2 points]** What is $\lfloor -\sqrt{2} \rfloor$?
 - (c) **[3 points]** Is the ceiling function surjective? Explain why.
 - (d) **[3 points]** Is the floor function injective? Explain why.
2. Suppose a procedure can be completed in M different ways, where approach i itself can be completed in m_i ways. (So approach 1 can be completed in m_1 different ways, approach 2 can be completed in m_2 ways, etc.) If none of the ways overlap, prove using the sum rule (recall the sum rule only applies to 2 ways, not M ways) that there are

$$\sum_{i=1}^M m_i \tag{1}$$

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

$$\sum_{i=j}^k a_i = a_j + a_{j+1} + a_{j+2} + \dots + a_k. \tag{2}$$

3.
 - (a) **[6 points]** How many surjective functions are there from set A to set B if $|A| = n$ and $|B| = 2$? Recall that a function $f : A \rightarrow B$ is surjective iff $\forall b \in B \exists a \in A, f(a) = b$.
 - (b) **[3 points]** Give an example of a real-world function $f : A \rightarrow B$ where $|A| = n$ and $|B| = 2$. (In other words, describe A , B , and f .)
4. **[6 points]** Use counting rules to count how many undirected, unweighted graphs are there on the vertices $\{a, b, c, d\}$ such that vertex a has degree 0, or vertex a has degree 1.
5. Graph representations:

- (a) **[9 points total]** (See grading rubric for pseudocode grading scheme.) Fill in the pseudocode for the following algorithms. You can assume that the graph is not directed, has no self-loops, and all edges have weight 1.

Algorithm 1: $\text{DegAdMat}(A, v)$

Input : Adjacency Matrix A of a graph $G = (V, E)$. A vertex $v \in V$.

Output: Degree of the vertex v .

Algorithm 2: $\text{DegAdList}(A, v)$

Input : Adjacency List A of a graph $G = (V, E)$. A vertex $v \in V$.

Output: Degree of the vertex v .

Algorithm 3: $\text{EdgeAdMat}(A, u, v)$

Input : Adjacency Matrix A of a graph $G = (V, E)$. Vertices $u, v \in V$.

Output: 1 if edge between u and v , zero otherwise.

Algorithm 4: $\text{EdgeAdList}(A, u, v)$

Input : Adjacency List A of a graph $G = (V, E)$. Vertices $u, v \in V$.

Output: 1 if edge between u and v , zero otherwise.

6. The input and output of a textual compression algorithm can be represented by a function $f : L \rightarrow L$, where L is the set of all strings. f takes in a string $s \in L$, and maps it to a new string s' that is ideally shorter than the original string. A lossless compression algorithm f is an algorithm where there exists a function $d_f : L \rightarrow L$ such that $d_f(f(s)) = s$ for all $s \in L$. (Most real world compression algorithms are not lossless.)
- (a) **[6 points]** What property (surjective, injective, bijective), should f have for it to be lossless? Why?
- (b) **[11 points]** Prove that if a lossless compression algorithm makes at least one input string smaller, then it must also make at least one input string larger. (See final page for hint.)

7. How long did you spend on this homework?

Hint: think pigeon hole principle