CS200 - Problem Set 7

1. (a) (Pseudocode grading parallels the regular code rubric.) Fill in the pseudocode for the following algorithms. You can assume that the graph is not directed, has no self-loops, and is unweighted. There is a triangle between three vertices if each of the three is connected to the other two.

Algorithm 1: `TriAdList`(A, u, v, w)
Input : Adjacency List A of a graph $G = (V, E)$. Vertices $u, v, w \in V$.
Output: 1 if there is a triangle among u v and w, zero otherwise.

Algorithm 2: `TriAdMat`(A, u, v, w)
Input : Adjacency Matrix A of a graph $G = (V, E)$. Vertices $u, v, w \in V$.
Output: 1 if there is a triangle amount u v and w, zero otherwise.

   (b) Create an expression for the worst case run-time for each function, using capital letters to represent constants, and summation notation to capture loops. You do not have to analyze this expression. NOTE: if you have a for loop that might end early, your expression should characterize what happens in the worst case.

2. The input and output of a textual compression algorithm can be represented by a function $f : L \to L$, 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. Let $L' = \{l \in L : (\exists s \in L : f(s) = l\}$.A lossless compression algorithm f is an algorithm where there exists a function $d_f : L' \to L$ such that $d_f(f(s)) = s$ for all $s \in L$. (Most real world compression algorithms are not lossless.)

   (a) What property (surjective, injective, bijective, neither), should f have for it to be lossless? In other words, for what functions f is it possible to create a perfect recovery function d_f? Why? (You do not need to come up with d_f, just state what property makes it possible to find one.)

   (b) 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.)

3. (a) Prove the following statement is false: $2^{2n} = O(2^n)$.

   (b) Provide C and k such that $2x^2 - 10 = \Omega(x)$

4. Determine how many bit strings of length 10 have

   (a) exactly three 0s?

   (b) at least seven 1s?

   (c) exactly three zeros or start with a 1?

   (Your answer may contain terms of the form $\binom{a}{b}$)

5. Consider the set of undirected, unweighted graphs on the vertex set $\{a, b, c, d\}$. If every graph is equal likely, what is the probability that vertex a has degree 0, or vertex a has degree 1? (Please calculate this probability by calculating the size of the sample space, and the size of the event.)

6. We study the following algorithm in CS302.

Algorithm 3: `dynamic`(n)

Input: An $n \times 3$ array L containing natural numbers.
$A, B \in \mathbb{N}$, a rectangular array Q of size $A \times B$ with all 0's

```
1  for k=1 to A do
2      for j=1 to B do
3          for q=1 to k-1 do
4              if Q[k,j] < Q[q,j] + Q[k-q,j] then
5                  Q[k,j] := Q[q,j] + Q[k-q,j];
6              end
7          end
8          for r=1 to j-1 do
9              if Q[k,j] < Q[k,r] + Q[k,j-r] then
10                 Q[k,j] := Q[q,j] + Q[k-q,j];
11             end
12         end
13         for i=1 to n do
14             if (k = L[i,1] and j = L[i,2]) or
                (k = L[i,2] and j = L[i,1]) then
15                 if Q[k,j] < L[i,3] then
16                     Q[k,j] := L[i,3];
17                 end
18             end
19         end
20     end
21 end
22 return Q[A,B];
```

Write an expression using summation notation and constants for the number operations used. (Do not analyze the expression.)

7. How long did you spend on this homework?

Hints for lossless proof

(a) Try creating a lossless function that just works on bit strings of length 3 or less to get some intutition for the proof. Using examples like this is not a proof, but might give you an idea for how to do the proof.

(b) Think pigeon hole principle