CS200 - Problem Set 8

1. **[11 points]** Suppose we are performing graph search on a graph $G = (V, E)$, starting at vertex $s \in V$. For the generic graph search algorithm we discussed in class, prove that for a vertex $v \in V$, there is a path from s to v if and only if v is explored by the search algorithm. (Recall a path is a sequence of connected edges.) See hints at end for my proof approaches.

2. Big-O

   (a) **[11 points]** Prove that $5 - 10x + 2x^2 = O(x^2)$.

   (b) **[11 points]** Prove that $\log_3(n^2) = O(\log_2(n))$. (This is a good question to review properties of logarithms, which come up often in computer science. Hints: recall if $\log_b(c) = x$ this means $b^x = c$. As a consequence, $a = b^{\log_b(a)}$. Also, $\log_b(a \times c) = \log_b(a) + \log_b(c)$. To prove this result, try to change the base of the term $\log_3(n^2)$ from 3 to 2. If you are feeling uncomfortable with this problem, go online and find extra practices problems dealing with logarithms and exponentiation.)

   (c) **[11 points]** Prove the following statement is false: $2^{2n} = O(2^n)$. (Hint: try a proof by contradiction!)

3. **[6 points]** Last problem set, you created pseudocode for 4 different graph algorithms: `DegAdMat`, `DegAdList`, `EdgeAdMat`, and `EdgeAdList`. The algorithms either find the degree of a vertex, or determine whether there is an edge between two vertices, but do so either using an adjacency list input, or an adjacency matrix input.

   Use big-O notation to bound the worst-case time complexity for each of your algorithms. Briefly explain, and comment on when it is good to use an adjacency matrix vs. an adjacency list.

4. We study the following algorithm in CS302.

Algorithm 1: `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

```
for k=1 to A do
    for j=1 to B do
        for q=1 to k-1 do
            if Q[k, j] < Q[q, j] + Q[k − q, j] then
                Q[k, j] := Q[q, j] + Q[k − q, j];
            end
        end
        for r=1 to j-1 do
            if Q[k, j] < Q[k, r] + Q[k, j − r] then
                Q[k, j] := Q[q, j] + Q[k − q, j];
            end
        end
        for i=1 to n do
            if (k = L[i, 1] and j = L[i, 2]) or
              (k = L[i, 2] and j = L[i, 1]) then
                if Q[k, j] < L[i, 3] then
                    Q[k, j] := L[i, 3];
                end
            end
        end
    end
end
return Q[A, B];
```

(a) **[3 points]** Write an expression using summation notation for the number operations used. (Do not analyze the expression)

(b) **[6 points]** Analyze the expression from part a by simplifying the sums, and then create an asymptotic bound for your simplified expression. (Your bound should be a sum of terms of the form $n^x A^y B^z$ for x, y, z non-negative integers.)

(c) **[6 points]** Explain how you can bound the asymptotic runtime without using summation notation, but instead using a worst-case analysis.

5. Determine how many bit strings of length 10 (an example of a bit string of length 10 is: 0101110010) have the following properties. See last page for a hint.

(a) **[6 points]** exactly three 0s?

(b) **[6 points]** at least seven 1s?

(c) **[6 points]** exactly three zeros or start with a 1?

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

6. [**6 points each**] Consider the following relations on the set of all humans. For each relation, explain whether or not the relation is reflexive, symmetric, and transitive. If it is an equivalence relation, what are the equivalence classes corresponding to the relation?

   (a) $R = \{(a, b) :\ a$ and b have a common grandparent$\}$.

   (b) $R = \{(a, b) :\ a$ and b have the same biological parents$\}$.

   (c) $R = \{(a, b) :\ a$ is the same height or taller than $b.\}$

7. How long did you spend on this homework?

- Problem 1: Forward: contradiction. Backwards: contrapositive.
- Problem 5: In these problems, one of the "tasks" you need to think about is choosing which locations out of the 10 possible locations to put 0's or 1's. Think about whether that is a $C(n, k)$ type of counting or a $P(n, k)$ type of counting.