COMP 1805 Discrete Structures I
Assignment 4

Due: August 2nd 2016, at the end of class

- Write down your name and student number on **every** page. The pages must be **stapled** together.
- You must have a cover page that clearly states **your name, student number, and course number**. If you do not have a cover page with this information, your assignment will not be marked.
- The questions should be answered in order.
- Every part of every question is worth 2 marks. The grading scheme is 2 points for a correct answer, 0 for a completely incorrect answer, and 1 point for something in-between.

Recall that, given two functions $f, g : \mathbb{R} \mapsto \mathbb{R}$, we say that f is $O(g(n))$ if

$$\exists c, n_0 \in \mathbb{R}^+ \left(\forall n \geq n_0 \left(f(n) \leq c \cdot g(n)\right)\right).$$

Similarly, we say that f is $\Omega(g(n))$ if

$$\exists c, n_0 \in \mathbb{R}^+ \left(\forall n \geq n_0 \left(f(n) \geq c \cdot g(n)\right)\right).$$

Finally, we say that f is $\Theta(g(n))$ if it is both $O(g(n))$ and $\Omega(g(n))$.

1. Suppose that an algorithm uses $7n^3 + 4^n$ elementary operations to solve a problem of size n. Suppose that your machine can perform one bit operation in 10^{-9} seconds, how long does it take your algorithm to solve a problem of size given below. Note, if your algorithm takes more than 60 seconds, answer in minutes. For more than 60 minutes, answer in hours. For more than 24 hours, answer in days. For more than 365 days, answer in years. For more than 100 years, answer in centuries!
   (a) 10
   (b) 20
   (c) 50
2. Prove that 4^n is *not* $O(2^n)$. You may assume that $c \geq n_0$.
3. Analyse the running time of the following algorithm in big-O notation. You may use either the exact summation method, or the rules of thumb.

Algorithm 1: FLOYDWARSHALL

Input: An undirected weighted graph $G = (V, E)$ with n vertices ($|V| = n$).
Output: An $n \times n$ matrix containing the length of the shortest path between each pair of vertices.

$D =$ a $n \times n$ matrix of minimum distances initialized to ∞
for *each vertex* v **do**
  $D[v][v] = 0$
for *each edge* (u, v) **do**
  $D[u][v] = D[v][u] = w(u, v)$ ($w(u, v)$ is the weight of edge (u, v))
for $k = 1$ **to** n **do**
  for $i = 1$ **to** n **do**
    for $j = 1$ **to** n **do**
      if $D[i][j] > D[i][k] + D[k][j]$ **then**
        $D[i][j] = D[i][k] + D[k][j]$
return D

4. Prove by induction that $\sum_{i=1}^{n} \frac{3}{4^i} = 1 - \frac{1}{4^n}$ for all $n \geq 2$.

5. This question is about tiling shapes with *L-trominoes*: three 1×1 squares attached so that they form an L-shape. To *tile* a shape means to fill it completely with non-overlapping copies of a base shape. For example, Figure 1 shows that we can tile an 8×8 square whose top-left corner is missing with L-trominoes.

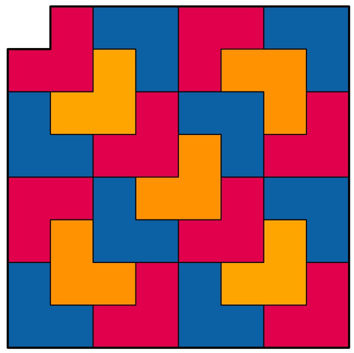

Figure 1: A L-trominoe tiling of an 8×8 square with its top-left corner missing.

   (a) Prove that for $n \geq 1$, any $2^n \times 2^n$ square with one corner missing can be tiled with L-trominoes.

   (b) What if the missing square is not a corner? Can we still tile all $2^n \times 2^n$ squares with a missing tile anywhere? If so, prove it. If not, give a counter-example: an $2^n \times 2^n$ square with a non-corner tile missing that cannot be tiled by L-trominoes, and explain why it cannot be tiled.

6. Give a recursive definition of the following functions, starting at $n = 1$.

   (a) $f(n) = 2n^2 - 5$

   (b) $f(n) = 4 \cdot 3^n$

   (c) $f(n) = 2^n + n^2$

7. Let the sequence $F_0, F_1, \ldots$ be defined by $F_0 = 0$, $F_1 = 1$, and $F_{n+2} = F_{n+1} + F_n$.

   Use induction to prove that $\sum_{i=1}^{n} F_{2i} = F_{2n+1} - 1$.

8. Draw the following graphs.

   (a) An undirected graph with the following adjacency matrix:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

   (b) A directed graph with the following adjacency list:

   1: 3
   2: 1, 5
   3: 5, 7
   4: 1, 3, 6
   5:
   6: 2
   7: 5, 6

9. Draw graphs with the following properties. Your graph should have at least 2 and at most 6 vertices.

   (a) A disconnected graph.

   (b) A connected bipartite graph.

   (c) A non-bipartite cycle.