**CPSC 221**
Theory Assignment 1

Brian Taylor
52311859
z2b0b
Lab L0D

1. Write a recursive algorithm (step-by-step descriptions or pseudocode, not C++ code, Java, or Racket) that returns the sum of every element of an array with a value less than x.

```
conditionalSum(array, x)
    if array length is zero return 0
    else
        if first element is less than x
            return first element + conditionalSum(rest of array, x)
        else
            return conditionalSum(rest of array, x)
```

2. The Traveling Salesman problem (TSP) is a famous problem for which there is no known, tractable solution (though efficient, approximate solutions exist). Given a list of cities and the distances in between, the task is to find the shortest possible tour (a closed walk in which all edges are distinct) that visits each city exactly once.

    Consider the following algorithm for solving the TSP:

```
n = number of cities
m = n x n matrix of distances between cities
min = (infinity)
for all possible tours, do:
    find the length of the tour
    if length < min
        min = length
        store tour
```

a) What is the complexity of this algorithm in terms of n (number of cities)? You may assume that matrix lookup is O(1), and that the body of the if-statement is also O(1). You need not count the if-statement or the for-loop guard (i.e., conditional) checks, etc., or any of the initializations at the start of the algorithm. Clearly show the justification for your answer.

The main question is how many possible tours there are. Since there are n cities and a tour is comprised of a list of all cities, appearing exactly once, there must be n! tours since the number of ways of ordering n objects is n!. So, the for-loop executes n! times. Finding the length of each tour requires adding the lengths of n edges (which would be O(n)). So, ignoring the matrix lookup and the body of the if-statement, overall the algorithm is O(n*n!).

b) Given your complexity analysis, assume that the time required for the algorithm to run when n=10 is 1 second. Calculate the time required for n=20 and show your work.

The time required for n=20, given that n=10 takes 1 second will be:

$$\frac{20(20!)}{10(10!)} \textit{ seconds}$$
$$= 2 \cdot 20 \cdot 19 \cdot 18 \cdot 17 \cdot 16 \cdot 15 \cdot 14 \cdot 13 \cdot 12 \cdot 11 \textit{ seconds}$$
$$= 1{,}340{,}885{,}145{,}600 \textit{ seconds}$$
$$\cong 42{,}491 \textit{ years}$$

3. Use the definition of big-Ω that we have discussed in class to show the following (be sure to provide appropriate values for c and $n_0$):

$$\textstyle\sum_{k=3}^{n}(k^2 - 2k) \quad \text{is } \Omega(n^3)$$

$$\sum_{k=3}^{n}(k^2 - 2k)$$
$$= \sum_{k=3}^{n} k^2 - 2\sum_{k=3}^{n} k$$
$$= \frac{n(n+1)(2n+1)}{6} - 5 - 2\left(\frac{n(n+1)}{2} - 3\right)$$
$$= \frac{2n^3 + 3n^2 + n - 6n^2 - 6n + 6}{6}$$
$$= \frac{n^3}{3} - \frac{n^2}{2} - \frac{5n}{6} + 1$$

Show that this is $\Omega(n^3)$:

if we let $c = \frac{1}{4}$ then $cn^3 \le \frac{n^3}{3} - \frac{n^2}{2} - \frac{5n}{6} + 1$ for all values of $n > n_0 = 8$

Check for n=8:

$$\frac{8^3}{4} \le \frac{8^3}{3} - \frac{8^2}{2} - \frac{5 \cdot 8}{6} + 1$$
$$128 \le 133$$

$$\lim_{n\to\infty} \frac{\frac{1}{4}n^3}{\frac{n^3}{3} - \frac{n^2}{2} - \frac{5n}{6} + 1}$$
$$= \lim_{n\to\infty} \frac{3n^3}{4n^3 - 6n^2 - 10n + 12}$$
$$= \lim_{n\to\infty} \frac{3}{4 - \frac{6}{n} - \frac{10}{n^2} + \frac{12}{n^3}}$$
$$= \frac{3}{4}$$

4. Suppose an algorithm solves a problem of size $x$ in at most the number of steps listed in each question below. Calculate the asymptotic time complexity (big-Θ or big-theta) for each example below. Show your work, including values for c and $x_0$ along the way.

<u>Note to marker:</u> I wasn't sure if it was acceptable to use $n$ when stating the asymptotic time complexity for each example given that the functions are functions of $x$, but since that seems to be the standard that's what I used.

a) $T(x) = 1$

Show that this is $O(1)$:

$$let\ c = 2{:}\ 2 \cdot 1 \geq 1\ \forall x > x_0 = 1$$

Show that this is $\Omega(1)$:

$$let\ c = 0.5{:}\ \ 0.5 \cdot 1\ \leq 1\ \forall x > x_0 = 1$$

Therefore $T(x) =\ 1\ is\ in\ \Theta(1)$.

b) $T(x) = 5x - 2$

Show that this is $O(n)$:

$$let\ c = 5{:}\ 5x \geq 5x - 2\ \forall x > x_0 = 1,\ \ by\ observation$$

Show that this is $\Omega(n)$:

$$let\ c = 4{:}\ \ 4x\ \leq 5x - 2\ \forall x > x_0 = 2$$

$$\lim_{n\to\infty} \frac{4x}{5x - 2}$$
$$= \lim_{n\to\infty} \frac{4}{5 - \frac{2}{x}}$$
$$= \frac{4}{5}$$

Therefore $T(x) =\ 5x - 2\ is\ in\ \Theta(n)$.

c) $T(x) = 3x^3 + 2 + x^2$

Show that this is $O(n^3)$:

$$let\ c = 4{:}\ 4x^3 \geq 3x^3 + 2 + x^2\ \forall x > x_0 = 2$$

$$\lim_{n\to\infty} \frac{4x^3}{3x^3 + 2 + x^2}$$
$$= \lim_{n\to\infty} \frac{4}{3 + \frac{2}{x^3} + \frac{1}{x}}$$
$$= \frac{4}{3}$$

Show that this is $\Omega(n^3)$:

$$let\ c = 3{:}\ \ 3x^3\ \leq 3x^3 + 2 + x^2\ \forall x > x_0 = 1, \qquad since\ 2 + x^2 > 0, \forall x$$

Therefore $T(x) =\ 3x^3 + 2 + x^2\ is\ in\ \Theta(n^3)$.

d) $T(x) = \log(x * 2x!)$

For this question I am assuming that 'log' is the natural logarithm. If it were instead the logarithm base 10, or some other base, the following would have to be adjusted, but only in terms of constants I believe.

A bit of transformation first:

$$\begin{aligned} &log(x \cdot x!) = log(x) + log(x!) \\ &= log(x) + log(x) + log(x-1) + \cdots + log(2) + log(1) \\ &\leq log(x) + log(x) + log(x) + \cdots + log(x) + 0 = xlog(x) \end{aligned}$$

Show that this is $O\big(nlog(n)\big)$:

$$let\ c = 1: xlog(x) \geq log(x \cdot x!)\ \forall x > x_0 = 2 \quad (see\ above)$$

Show that this is $\Omega(nlog(n))$:

$$\begin{aligned} &log(x!) \cong xlog(x) - x \quad (by\ Stirling's\ Approximation) \\ &log(x \cdot x!) \cong log(x) + xlog(x) - x \end{aligned}$$

$$let\ c = 0.5:\ 0.5xlog(x) \leq log(x \cdot x!)\ \forall x > x_0 = 4$$

$$\begin{aligned} &0.5xlog(x) \leq xlog(x) + log(x) - x\ \ \forall x > x_0 = 4 \\ &xlog(x) \leq 2xlog(x) + 2log(x) - 2x\ \ \forall x > x_0 = 4 \\ &0 \leq xlog(x) + 2log(x) - 2x\ \forall x > x_0 = 4 \end{aligned}$$

$$\begin{aligned} &\lim_{n \to \infty} \frac{\frac{1}{2}xlog(x)}{xlog(x) + \log(x) - x} \\ &= \lim_{n \to \infty} \frac{xlog(x)}{2xlog(x) + 2\log(x) - 2x} \\ &= \lim_{n \to \infty} \frac{1}{2 + \frac{2}{x} - \frac{2}{\log(x)}} \\ &= \frac{1}{2} \end{aligned}$$

Therefore $T(x) = \ log(x \cdot x!)\ is\ in\ \Theta(nlog(n))$.