***Bi Nguyen***
***913875445***
***Section: A05***

# ECS 32B: Fall 2019 Homework Assignment 1

**Due Date:** No later than Tuesday, January 22, 9:00pm

**0.** Make sure to clearly identify your name, student number, and your discussion section on the work you submit. Submit only one pdf file via Canvas...do not submit a separate file for each problem.

**1.** 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:

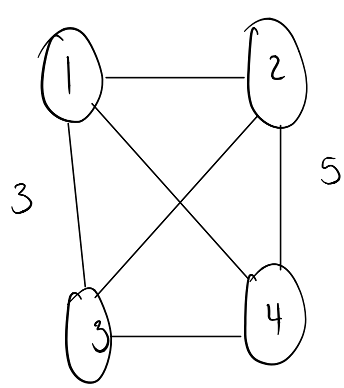
*n = number of cities*
*m = n x n matrix of distances between cities min = (infinity)*
*for all possible tours, do:*

*find the length of the tour if length < min*

*min = length*

*store tour*

a) What is the worst-case (big-O) 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 conditional (i.e., testing to see when the for-loop is done), or any of the initializations at the start of the algorithm. Clearly show the justification for your answer.




| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 4 | 3 | 7 | 6 |
| 2 | 4 | 0 | 1 | 2 | 7 |
| 3 | 3 | . | 0 | . . . | . . |
| 4 | 7 | . . . | . | 0 | . . |
| 5 | 6 | . . | . . | . . | 0 |

Considering that the sales man has to go through each city once per trip and return to the city that he started. So the salesman has to go from 1→2, 2→1, then 1→2→3 then 1→3→2 then 2→1→3 then 2→3→1, 3→1→2, 3→2→1 which is equivalent to 3! and so the complexity of all the possible distance would be n!

And storing the distance will take n times since it's calculating the distance each complete trip. So the worst case scenerio 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.

since O(n n!) then

$$\frac{20 \cdot 20!}{10 \cdot 10!} = \frac{x}{1} = 1.340885176 \text{ E}13$$

**2.** Suppose an algorithm solves a problem of size $n$ in at most the number of steps listed for each T($n$) given below. Calculate the asymptotic time complexity (big-Θ or big-theta) for each T($n$). Show your work, including values for $c$ and $n_0$ along the way.

a) T($n$) = 1

Big-O

$T(n) \leq O(1)$

$1 \leq c$

Big Ω

$T(n) \geq O(1)$

$1 \geq C$

b) T($n$) = 5$n$ – 2

Big-O

let c = 5

let $n_0 \geq 1$

$T(n) \leq O(n)$

$5n-2 \leq cn$

$\frac{5n-2}{5} \leq \frac{5n}{5}$

$n - \frac{2}{5} \leq n$

$1 - \frac{2}{5} \leq 1$

Big-Ω

let c=1

let $n_0 \geq 1$

$T(n) \geq \Omega(n)$

$5n-2 \geq cn$

$5n \geq n+2$

$n \geq \frac{n}{5} + \frac{2}{5}$

$1 \geq \frac{1}{5} + \frac{2}{5}$

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

Big-O

let c = 6

let $n_0 \geq 1$

$T(n) \leq O(n^3)$

$3n^3 + 2 + n^2 \leq cn^3$

$\frac{3n^3}{6} + \frac{n^2}{6} + \frac{2}{6} \leq n^3$

$\frac{3}{6} + \frac{1}{6} + \frac{2}{6} = 1 \leq 1$

Big-Ω

let c=1

let $n_0 \geq 1$

$T(n) \geq O(n^3)$

$3n^3 + 2 + n^2 \geq cn^3$

$3n^3 \geq cn^3 - n^2 - 2$

$n^3 \geq \frac{n^3 - n^2 - 2}{3}$

$1 \geq \frac{1-1-2}{3}$

d) T($n$) = log ($n$ * 2$n$!)

$\log(n \cdot 2n!)$

$\hookrightarrow \log(n!) = \log(n \cdot (n-1)(n-2) \ldots)$

$= \underbrace{\log(n) + \log(n-1) + \log(n-2) \ldots}_{\text{each takes } \log n \text{ thus up to } n}$

$= n\log(n)$

Big - O

$T(n) \le O(n\log(n))$

let $n_0 = 1$

let $c \geq 1$

$\log(n \cdot 2n!) \le cn\log(n)$

$\log(n \cdot 2n!) \le cn\log n$

$\log n + \log 2 + \log n! \le cn\log n$

$$\frac{\log n + \cancel{\log_2 2}^{\,1} + \log n!}{n\log n} \le \frac{cn\log n}{n\log n}$$

$\log n - n\log n + \overset{1}{\log_2 2} - \log n! + n\log n \le C$

$0 - 0 + 1 - 0 + 0 \le C$

$1 \le C$

Big $\Omega$

$T(n) \geq O(n\log n)$

let $c = 1$

let $n = 1$

$\log(n \cdot 2n!) \geq c(n\log n)$

$\log n + \log_2 2 + \log n! \geq cn\log n$

$$\frac{\log n + \log n! + \log_2 2}{n\log n} \geq \frac{cn\log n}{n\log n}$$

$\log n - n\log n + \log n! + 1 \geq C$

$0 - 0 + 0 - 0 + 1 \geq C$

$1 \geq 1$

properties

$\log_b b = 1$

$\log 1 = 0$

$\frac{\log n}{n\log n} = \log n - n\log n \ldots$