Kunal Patel

# CSE 310 Assignment #2

**(Max. Points: 30)**

**Due on: Monday, Sep. 23, 2019, 11:59pm Arizona time**

**General Instructions:**

- This is an individual assignment, please do not collaborate. If there's programming part, make sure that you write every line of your own code. Using code written by someone else will be considered a violation of the academic integrity and will result in a report sent to the Dean's office.
- For all written exercises: **your answer should be clearly typed or written and must be saved in .pdf or .jpg format**. Note: **unreadable answer receives no credits!**
- All assignments must be submitted through the link posted on Canvas, we do NOT accept any hand-in submissions or submissions sent through emails!
- Submission link will be closed automatically once the due date/time is past and **no late assignment will be accepted**.
- You will be allowed multiple times to submit the assignment before the due date/time, but we will only grade your last submission.

**Objectives**

- Asymptotic notation.
- MergeSort, QuickSort, recurrence, Master method, etc.
- Be familiar with OpenMP

**Questions**

1. [1 pt each, total 3 pts] For each of the following pairs of functions $f(n)$ and $g(n)$, determine only one of $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, or $f(n) = \Theta(g(n))$. Note: if $f(n) = \Theta(g(n))$, then do not choose $f(n) = O(g(n))$ or $f(n) = \Omega(g(n))$.

1) $f(n) = 85n^2\sqrt{n} + 3n\sqrt{n}$, $g(n) = 21n^2\sqrt{n} + 87n\sqrt{n} + 3n^3$

$f(n) = O(g(n))$

2) $f(n) = (15n^3 + 4n^2)/3$ , $g(n) = 8n^4$

$f(n) = O(g(n))$

3) $f(n) = log_3(n^8)$, $g(n) = log_3(n^5)$

$f(n) = \Theta(g(n))$

2. [2 pts] Suppose that the running time of an algorithm $A$ is $20n^4$, and the running time of an algorithm $B$ is $1500n^2$. What is the largest value of $n$ (a positive integer) for which the running time of the algorithm $A$ is smaller than that of $B$?

where $n=8$, $1500n^2 > 20n^4$

$1500n^2 = 20n^4$

$n \approx 8.7$ => round down gives smallest integer where $1500n^2 > 20n^4$

3. [4 pts] Suppose that $T(n) = 3$ for $n = 1$, and for all $n \geq 2$, $T(n) = T(n-1) + 3n - 2$. Solve this recurrence exactly by drawing a ***recursion tree***. (You will need to give an exact explicit solution for $T$). $T(1) = 3$, $T(n) = T(n-1) + 3n - 2$

| $T(n)$ | Total | Level |
|---|---|---|
| $3n-2$ | $3n-2$ | 0 |
| $+ T(n-1)$ | | |
| $3n-5$ | $3n-5$ | 1 |
| $+ T(n-2)$ | | |
| $3n-8$ | $3n-8$ | 2 |
| $T(1)$ | 3 | $n-1$ |

$\sum_{i=0}^{n-1} (3n-2+3i)$ $\longrightarrow \sum_{i=0}^{n-1} 3n + \sum_{i=0}^{n-1} -2 + \sum_{i=1}^{n-1} 3i$

$= n(3n) + n(-2) - \frac{3}{2}n^2 + \frac{3}{2}n$

$= 3n^2 + \frac{n}{2}$

$= O(n^2)$

4. [3 pts each, total 9 pts] Use the master method to give tight asymptotic bounds (θ bound) for the following recurrences. Specify clearly *a, b, f(n)* or ε value and which case you applied.

*1)* $T(n) = 2T(n/4) + 1$

case 1:
$f(n) = n^0 = 1 \Rightarrow c = 0$
$\log_4 2 = 1/2 \Rightarrow c < \log_b a$
$\Rightarrow T(n) = \theta(n^{1/2})$

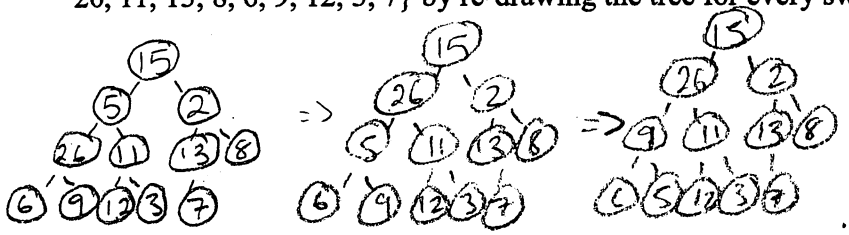
*2)* $T(n) = 4T(n/2) + n^2$

case 2:
$f(n) = n^2 \Rightarrow c = 2$
$\log_2 4 = 2 \Rightarrow c = \log_b a$
$\Rightarrow T(n) = \theta(n^2 \log n)$

*3)* $T(n) = 3T(n/5) + n^2$

case 3:
$f(n) = n^2 \Rightarrow c = 2$
$\log_5 3 = 0.68 \Rightarrow c > \log_b a$
$\Rightarrow T(n) = \theta(n^2)$

5. [4 pts] Illustrate the operation of MAX-HEAPIFY(*A*, 2) on the array A = { 15, 5, 2, 26, 11, 13, 8, 6, 9, 12, 3, 7} by re-drawing the tree for every swap.



6. [Total 8 pts] Read the notes on OpenMP (file *OpenMP.pdf*) and compile/execute the sample programs included in the zip file (file *openmp.zip*) as you go to familiarize yourself with the various OpenMP directives. Answer the following questions; it is expected that you will run each program multiple times. You may need to modify the program(s) to answer the questions. Be sure to explain the experiments you conducted.

1) [3 pts] For the program in the file *parfor.cc* vary the value of the variable *n* and the number of threads specified in the num threads clause. How are the iterations distributed among threads? Be sure to try out fewer iterations than threads, and more iterations than threads, etc. Write your observation.

- Iterations increase in correspondence of threads
- Increasing thread in code makes no differenc, maximum of 4 threads on server.

Ex: thread 0 , iteration 0-2
thread 1 , iteration 3-5 // continues from where left off

2) [3 pts] For the program in the file *private.cc* explore the values of the variables *a* and *i* before, after, and inside the parallel region. Try initializing the variables before the *#pragma* and also not initializing them. Write your observation.

**Before**
- a = 55, i = 107
- iterations for threads work the same as prev. prob
- a and i values came from local vars

**After**
- a = 55, i = 107
- only because declared before print, otherwise 0
- Iterations for threads work the same

**Inside**
- a = 0, i = 0
- In iterations values of a and i are the ones declared so a = 110, i = 107
- Iterations for threads work the same

3) [2 pts] For the program in the file *reduction.cc* explore the run time of the reduction clause by varying the number of threads in the num threads clause Write your observation.

Generally increasing n took longer but varried sometimes was quicker or longer. I decided to take an average to judge the general trend.