CSE 310 Assignment #2

akshayms@asu.edu

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) = Ω(g(n)), or f(n) =Θ(g(n))*. Note: if *f(n)= Θ(g(n))*, then do not choose *f(n) = O(g(n))* or *f(n) = Ω(g(n))*.

1) $f(n) = 72n^2\sqrt{n} + 87n\sqrt{n} + 3n^3$, $g(n) = 15n^2\sqrt{n} + 3n\sqrt{n}$

f(n)/g(n) > C when n>=0
i.e., ***f(n) = Ω(g(n))***

2) $f(n) = 9n^5$, $g(n) = (26n^4 + 4n^3)/3$

f(n)>=g(n), n>=2
f(n)<=g(n), n>0

f(n)= Θ(g(n))

3) $f(n) = log_4(n^8)$, $g(n) = log_4(n^5)$

f(n)/g(n) = $log_4(n^8)/ log_4(n^5)$ = n^3

f(n) = Ω(g(n))

2. [2 pts] Suppose that the running time of an algorithm *A* is 1300n^2, and the running time of an algorithm *B* is 50n^4. What is the largest value of *n* (a positive integer) for which the running time of the algorithm *A* is larger than that of *B*?

$1300n^2 > 50n^4$

- ⇨ $26 n^2 > n^4$
- ⇨ $26 > n^2$
- ⇨ n<=5

The largest value n can take is 5

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

$$T(n) = \begin{cases} 3, & n = 1 \\ T(n-1) + 2n - 3, & n \geq 2 \end{cases}$$

	Cost	Level
$T(n) = 2n-3 + T(n-1)$	$2n-3$	0
$= 2(n-1) - 3 + T(n-2)$	$2n-5$	1
$2(n-2) - 3 + T(n-3)$	$2n-7$	2
$\vdots$	$\vdots$	

$(n-1)^{th}$ term $= 2n - (2n-1) = \underline{\underline{1}}$

Total Cost $= 2n-3 + 2n-5 + 2n-7 + \ldots + 2n-(2n-1)$

$= (n-1)(2n) - [3+5+7+\ldots+2n-1]$

$= \underline{\underline{2n^2 - 2n - [3+5+7+\ldots 2n-1]}}$

is a Quadratic equation of n^2

so, maximum running time $= \underline{\underline{\theta(n^2)}}$

4. [3 pts each, total 9 pts] Use the master method to give tight asymptotic bounds (θ bound) for the following recurrences. (Specify, *a, b, f(n)*, and ε value when they apply.)

1) $T(n) = 16T(n/4) + n$

$a = 16$, $b=4$, $f(n) = n$, $\varepsilon = 0$

$n^{\log_4 16} = n^2$ grows faster than $f(n)$

hence, Case #1 applies

$\mathbf{T(n) = \Theta(n^2)}$

2) $T(n) = 8T(n/2) + n^3$

$a = 8$, $b=2$, $f(n) = n^3$, $\varepsilon = 0$

$n^{\log_2 8} = n^3$ is same as $f(n)$

hence, Case #3 applies

$\mathbf{T(n) = \Theta(n^3 \log n)}$

3) $T(n) = 4T(n/8) + n^2$

$a = 4$, $b=8$, $f(n) = n^2$, $\varepsilon = 2$

when $\varepsilon = 2$

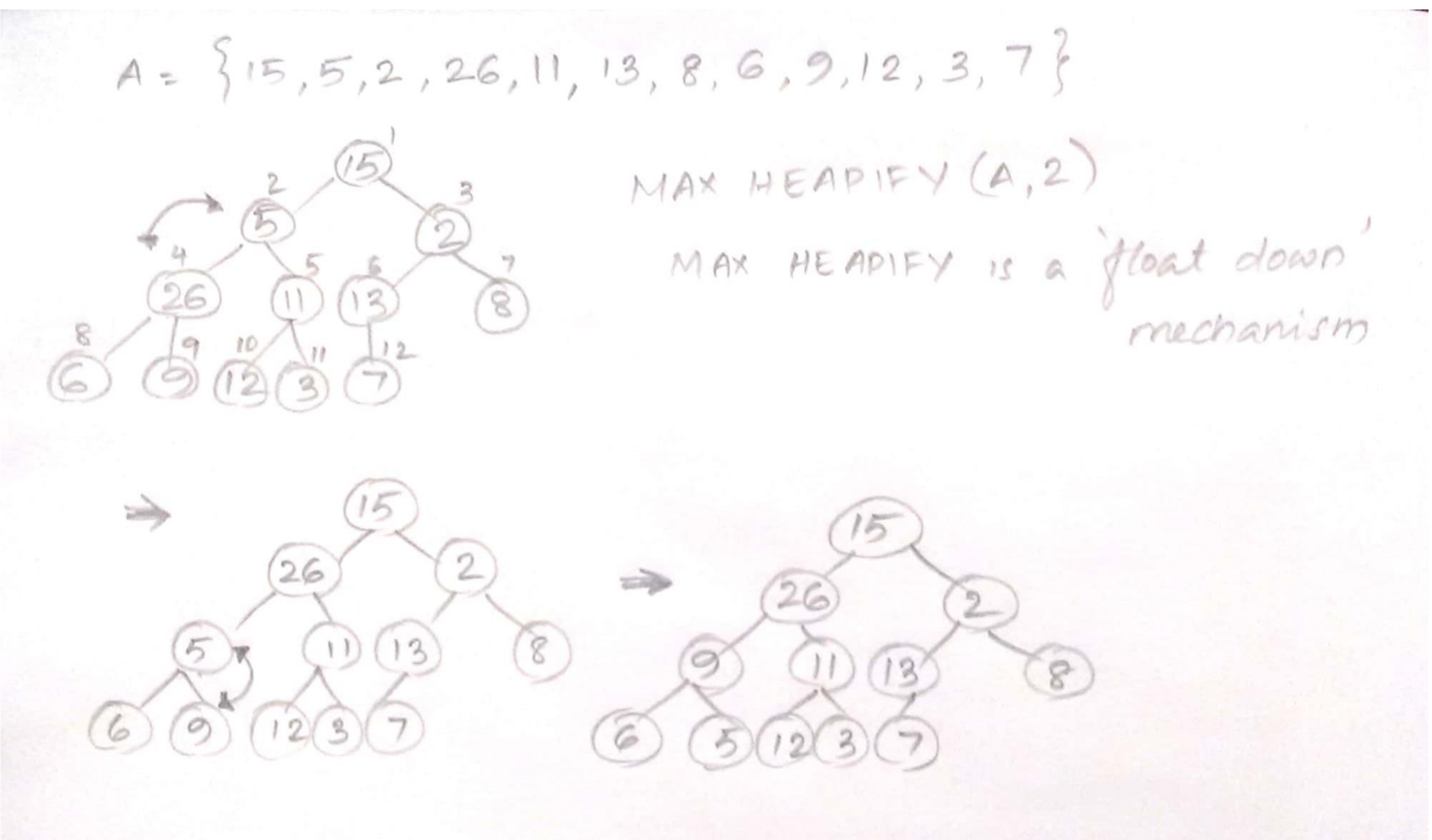
$n^{\log_8 2} = n^{1/3}$

regularity condition => $n^{1/3} < f(n)$

hence case #3 applies

$\mathbf{T(n) = \Theta(n^2)}$

5. 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.

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.

The iteration is distributed among the threads, the order of execution is arbitrary.
When the number of variables ‘n’ is larger than the number of threads, threads are given multiple executions for different ‘i’ values selected in an arbitrary fashion.
When the number of threads are larger than variables ‘n’ each thread gets to execute each iteration is arbitrary order.

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 not initializing them. Write your observation

The values initialized for ‘a’ and ‘i’ are preserved till the end of the program. Inside the parallel region the ‘a’ and ‘i’ are having different values for each thread but it doesn’t affect the variables declared outside the parallel region. Each thread is having a private copy of ‘a’ and ‘i; variables.
When ‘a’ and ‘i’ are not initialized arbitrary values are added to them and kept unchanged till the end of the program.

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

When the input size is lesser than the number of threads, parallel region executes faster. The other part follows more of a constant running time all the time for all kinds of inputs and for larger inputs than the number of threads this method has lower running time.