INSTITUTE OF TECHNOLOGY

TCSS 343 - Assignment 2

April 12, 2016

1 GUIDELINES

Homework should be electronically submitted to the instructor by midnight on the due date. A submission link is provided on the course Canvas Page. The submitted document should be typeset using any common software and submitted as a PDF. We strongly recommend using LaTeX to prepare your solution. You could use any LaTeX tools such as Overleaf, ShareLatex, TexShop etc. Scans of handwritten/hand drawn solutions are not acceptable.

Each problem is worth a total of 10 points. Solutions receiving 10 points must be correct (no errors or omissions), clear (stated in a precise and concise way), and have a well organized presentation. Show your work as partial points will be awarded to rough solutions or solutions that make partial progress toward a correct solution.

Remember to cite all sources you use other than the text, course material or your notes.

2 Problems

2.1 Understand

For this problem consider the problem of finding the maximum element in a list of integers.

```
Maximum Integer in a List (MAX)
  Input:  A[a...b] list of integers.
  Output:  A[i] for some a ≤ i ≤ b such that A[i] ≥ A[j] for all a ≤ j ≤ b.
```

Let $M(A[a \dots b])$ represent the output of the `MAX` problem on input $A[a \dots b]$. Let $\max(a, b)$ be a simple function that returns the maximum of two elements. Let $m = \left\lfloor \frac{a+b}{2} \right\rfloor$ be the midpoint between a and b, let $t_1 = \left\lfloor \frac{a+b}{3} \right\rfloor$ be the point one third of the distance from a to b, and let $t_2 = \left\lfloor \frac{2(a+b)}{3} \right\rfloor$ be the point two thirds of the distance from a to b.

(3 points) 1. Below is a self-reduction for the `MAX` problem. State a recursive algorithm using pseudocode for finding the maximum element based on this self-reduction.

$$M(A[a \dots b]) = \begin{cases} A[a] & \text{if } a = b \\ \max(M(A[a \dots b-1]), A[b]) & \text{if } a < b \end{cases}$$

(2 points) 2. Using the same reduction as part 1 now state a recurrence $T(n)$ that expresses the worst case run time of the recursive algorithm. Find a similar recurrence in your notes and state the tight bound on $T(n)$.

(3 points) 3. Below is a self-reduction for the `MAX` problem. State a recursive algorithm using pseudocode for finding the maximum element based on this self-reduction.

$$M(A[a \dots b]) = \begin{cases} -\infty & \text{if } a > b \\ A[a] & \text{if } a = b \\ \max(\max(M(A[a \dots t_1]), M(A[t_1+1 \dots t_2])), M(A[t_2+1 \dots b])) & \text{if } a < b \end{cases}$$

(2 points) 4. Using the same reduction as part 3 now state a recurrence $T(n)$ that expresses the worst case run time of the recursive algorithm. You do not need to formally prove your recurrence, but you have to show that it is a reasonable guess by using a recursion tree or by the repeated substitution method. *Hint: assume that n is a power of 3.*

Grading You will be docked points for errors in your math, disorganization, unclarity, or incomplete proofs.

2.2 Explore

For this problem consider the problem of finding the product of a list of integers.

Product of All Integers in a List (PROD)
  Input: $A[a \dots b]$ list of integers.
  Output: $p = \prod_{i=a}^{b} A[i]$.

Let $P(A[a \dots b])$ represent the output of the PROD problem on input $A[a \dots b]$ (by convention, the product of an empty list is defined to be 1).

(4 points) 1. State two different divide-and-conquer self-reductions for the PROD problem.

(4 points) 2. Give recursive algorithms based on your divide-and-conquer self-reductions to solve the PROD problem.

(2 points) 3. What are the worst-case runtimes of the solutions you have obtained? (Just state the runtimes. You do not need to show your work.)

2.3 Expand

Consider the following recurrence $T(n)$:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ 2T(\lfloor \frac{n}{4} \rfloor) + 16 & \text{if } n > 1 \end{cases}$$

(3 points) 1. Use the recursion tree or repeated substitution method to come up with a good guess for a bound $g(n)$ on the recurrence $T(n)$.

(2 points) 2. State and prove by induction a theorem showing $T(n) \in \Theta(g(n))$.

Consider the following recurrence $T(n)$:

$$T(n) = \begin{cases} c & \text{if } n = 1 \\ T(\lfloor \frac{n}{2} \rfloor) + T(\lfloor \frac{n}{4} \rfloor) + 4n & \text{if } n > 1 \end{cases}$$

(1 point) 3. Draw the first six levels of the recursion tree by drawing all recursive calls of the same size on the same level. Make sure on each level you indicate the size of the recursive call and the number of recursive calls.

(1 point) 4. Express the cost of all levels of the recursion tree as a sum over the cost of each level of the recursion tree.

(1 point) 5. Give a function $g(n)$ and show it is an upper bound on the sum.

(2 points) 6. State and prove by induction a theorem showing $T(n) \in \Theta(g(n))$.

Grading You will be docked points for errors in your math, disorganization, unclarity, or incomplete proofs.

2.4 CHALLENGE

Consider the following observation on the randomized Quicksort algorithm. Because the pivot is chosen uniformly at random among the n input element, with probability $\frac{1}{2}$ the chosen pivot's position on the sorted list will be between $\frac{n}{4}$ and $\frac{3n}{4}$ (i.e. a good pivot), and with probability $\frac{1}{2}$ the chosen pivot's position on the sorted list will be between 1 and $\frac{n}{4}$ or between $\frac{3n}{4}$ and n (i.e. a bad pivot).

(1 point) 1. State a recurrence that expresses the worst case for bad pivots.

(1 point) 2. State a recurrence that expresses the worst case for good pivots.

(1 point) 3. State a recurrence that expresses the expected worst case by combining the first two recurrences.

(2 points) 4. Prove by induction that your recurrence is $\Theta(n \log n)$.

Grading Correctness and precision are of utmost importance. Use formal proof structure for the Big-Theta bounds. You will be docked points for errors in your math, disorganization, unclarity, or incomplete proofs.