INSTITUTE OF TECHNOLOGY

TCSS 343 - Assignment 3

Version 1.0 – Zhihao Yang

May 29, 2018

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 acceptable, but you will be docked 1 point per problem if the handwritting is unclear.

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

In this problem use the Master Theorem to find and prove tight bounds for these recurrences (4 points each).

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

   Answer:

   $a = 2, b = 4, n^{\log_b a} = n^{\log_4 2} = n^{\frac{1}{2}}$. Since $f(n) = 16 \in \theta(n^0)$.

   So this is case one, $f(n) \in O(n^{\frac{1}{2}-\epsilon})$ for $0 < \epsilon \le \frac{1}{2}$. Then $T(n) \in \theta(n^{\frac{1}{2}})$.

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

   $a = 4, b = 4, n^{\log_b a} = n^{\log_4 4} = n^1$. Since $f(n) = 16n \in \theta(n)$.

   So this is case two, $f(n) \in \theta(n)$. Then $T(n) \in \theta(n^1 \log n)$.

3. $$T(n) = \begin{cases} c & \text{if } n \le 1 \\ 8T\left(\lfloor \frac{n}{4} \rfloor\right) + 16n & \text{if } n > 1 \end{cases}$$

   $a = 8, b = 4, n^{\log_b a} = n^{\log_4 8} = n^2$. Since $f(n) = 16n \in \theta(n)$.

   So this is case one, $f(n) \in O(n^{2-\epsilon})$ for $0 < \epsilon \le 1$. Then $T(n) \in \theta(n^2)$.

4. $$T(n) = \begin{cases} c & \text{if } n \le 1 \\ 2T\left(\lfloor \frac{n}{3} \rfloor\right) + 16n & \text{if } n > 1 \end{cases}$$

   $a = 2, b = 3, n^{\log_b a} = n^{\log_3 2}$. Since $f(n) = 16n \in \theta(n)$.

   So this is case three, $f(n) \in \Omega(n^{\log_3 2+\epsilon})$. Check:

   $$af(\frac{n}{b}) \le df(n)$$
   $$2 \cdot 16 \cdot \frac{n}{3} \le d \cdot 16n$$
   $$\frac{2}{3} \cdot 16n \le d \cdot 16n$$
   $$\frac{2}{3} \le d$$

   we can pick $d = \frac{2}{3} < 1$ here to satisfy the requirement. Then $T(n) \in \theta(n)$.

5.

$$T(n) = \begin{cases} c & \text{if } n \leq 1 \\ 25T\left(\lfloor \frac{n}{3} \rfloor\right) + 16n^3 & \text{if } n > 1 \end{cases}$$

$a = 25, b = 3, n^{\log_b a} = n^{\log_3 25}$. Since $f(n) = 16n^3 \in \theta(n^3)$.

So this is case one, $f(n) \in O(n^{\log_3 25 - \epsilon})$ for $0 < \epsilon \leq \log_3 25 - 3$. Then $T(n) \in \theta(n^{\log_3 25})$.

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

2.2 EXPLORE

For this problem consider raising an integer a to the power n (another positive integer). Let a^n express the output to this problem.

1. (2 point) Express this problem formally with input and output conditions.

   Answer:

   Integer Exponentiation

   Input: an integer a, an positive integer n.

   Output: a^n the integer a to the power of n.

2. (4 points) Describe a simple brute force algorithm to compute this product. How many multiplications does it take asymptotically in the worst case?

   Answer:

   IntExp(a, n)

   Let k = 1;

   For i = 0 to n:

   k *= k;

   EndFor

   Return k;

   End IntExp;

3. (2 point) Assume for the moment that $n > 1$ and is even. Express the value of a^n as a product of two subproblems.

   Answer:

   $a^n = a^{\frac{n}{2}} \cdot a^{\frac{n}{2}} = (a^{\frac{n}{2}})^2$

4. (2 point) Now assume $n > 1$ and is odd. Express the value of a^n as a product of two subproblems (and possibly other terms).

   Answer:

   $a^n = a^{\frac{n-1}{2} + \frac{n-1}{2} + 1} = a^{\frac{n-1}{2}} \cdot a^{\frac{n-1}{2}} \cdot a = (a^{\frac{n-1}{2}})^2 \cdot a$

5. (4 points) State a self-reduction for your problem.

   Answer:

$$\text{IntExp}(a,n) = \begin{cases} 1 & \text{if } n = 0 \\ \text{a} & \text{if } n = 1 \\ \text{IntExp}(a,\frac{n}{2}) \cdot \text{IntExp}(a,\frac{n}{2}) & \text{if } n > 1 \text{ and n is even} \\ \text{IntExp}(a,\frac{n-1}{2}) \cdot \text{IntExp}(a,\frac{n-1}{2}) \cdot a & \text{if } n > 1 \text{ and n is odd} \end{cases}$$

6. (4 points) State a recursive algorithm that solves the problem.

   Answer:

   IntExp(a, n)

   if $n = 0$ return 1;

   if $n = 1$ return a;

   if $n > 1$ and n is even,

   temp = IntExp($a, \frac{n}{2}$);

   return temp*temp;

   if $n > 1$ and n is odd,

   temp = IntExp($a, \frac{n-1}{2}$);

   return temp*temp*a;

   End IntExp;

7. (2 point) State a tight asymptotic bound on the number of multiplications used by your algorithm in the worst case.

   If $n \leq 1$, there is no multiplications needed. If $n > 1$, you need one multiplication(or two multiplications for odd integers) for return and multiplications to divide a by 2 until it reaches the smallest integer greater than 1.

   So the dominant action should be $\Theta(\log n)$ for dividing by 2, then plus 2 multiplications. Hence, the worst case is $\Theta(\log n)$.

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

2.3 EXPAND

For this problem consider the problem of finding the median of integers in two sorted lists of size n. Let $M(A,B)$ express the output to this problem.

1. (2 point) Express this problem formally with input and output conditions.

   Answer:

   Find the median

Input: two sorted list, $A[1\dots n], B[1\dots n]$, with size of n

Output: A[i] or B[j], a integer which is the median in two sorted lists, both i, j should be greater than 0 and less than n.

2. (4 points) Describe a simple algorithm to compute the median. How many operations does it take asymptotically in the worst case?

   Answer:

   FindMedian($A[1\dots n], B[1\dots n]$)

   Let K be by a list of size $2n$;

   K = Combine(A, B);

   return $K[\lceil \frac{n}{2} \rceil]$;

   End FindMedian;

   The combine two lists will takes kn operations for some positive integer k and plus some constant operations. So the algorithm takes $O(n)$.

3. (6 points) Show how you can use the medians of the two lists to reduce this problem to its subproblems. State a precise self-reduction for your problem.

   Answer:

   a) First, the easiest case is that we only have one element in both lists. So we can just pick the smaller number as our median.

   b) Second, when the medians of both lists are exactly equal, our result should be just that median.

   c) Third, if we can't determine the median of two lists by just looking at the medians of both lists, for example $A[\lceil \frac{n}{2} \rceil] < B[\lceil \frac{n}{2} \rceil]$, we need to find a bigger number in list A and a smaller number in list B. So what we can do is separate both lists by a half. For list A, we need to pick the second half and the first half for list B, then pass them into the algorithm again.

   d) Finally, when $A[\lceil \frac{n}{2} \rceil] > B[\lceil \frac{n}{2} \rceil]$, we will do the same thing with the third case. We can separate both lists by a half. Now we should pick the first half of list A and second half for list B, then pass them into the algorithm again.

$$\text{FindMedian}(A[1\dots n], B[1\dots n]) = \begin{cases} \text{Min\{A[n], B[n]\}} & \text{if } n = 1 \\ A[\lceil \frac{n}{2} \rceil] & \text{if } n > 1 \text{ and } A[\lceil \frac{n}{2} \rceil] = B[\lceil \frac{n}{2} \rceil] \\ \text{FindMedian}(A[\lceil \frac{n}{2} \rceil \dots n], B[1\dots \lfloor \frac{n}{2} \rfloor]) & \text{if } n > 1 \text{ and } A[\lceil \frac{n}{2} \rceil] < B[\lceil \frac{n}{2} \rceil] \\ \text{FindMedian}(A[1\dots \lfloor \frac{n}{2} \rfloor], B[\lceil \frac{n}{2} \rceil \dots n]) & \text{if } n > 1 \text{ and } A[\lceil \frac{n}{2} \rceil] > B[\lceil \frac{n}{2} \rceil] \end{cases}$$

4. (6 points) State a recursive algorithm that solves the problem based on your reduction.

   Answer:

FindMedian($A[1\dots n, B[1\dots n]]$)

Set mid = $\lceil \frac{n}{2} \rceil$; Set midf = $\lfloor \frac{n}{2} \rfloor$;

if $n = 1$, return Min$\{A[1], B[1]\}$;

if $n > 1$ and A[mid] = B[mid], return A[mid];

if $n > 1$ and A[mid] < B[mid], FindMedian(A[mid...n], B[1...midf]);

if $n > 1$ and A[mid] > B[mid], FindMedian(A[1...midf], B[mid...n]);

End FindMedian

5. (2 point) State a tight asymptotic bound on the number of operations used by your algorithm in the worst case.

   Answer:

   For the four if statements above in part 4, two of them are the base cases and other two are mutually exclusive, which mean only one of them would get triggered everytime the algorithm gets called. Everytime calling the FindMedian(...) will count as one operation. So it's just going to be very similiar to a binary search.

   If $n \leq 1$, there will just be constant operations. If $n > 1$, it should be $\Theta(\log n)$ operations since the numbers of cutting both lists by half is going to be $\log n$.

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

2.4 CHALLENGE

Consider the problem of finding the closest pair of points out of a set of points in a 2-dimensional space. This problem is discussed at a high level in Section 5.5 of the text. This description can be imprecise at omit important details. In this problem you will fill in the details omitted and add precision.

1. (1 point) Express this problem formally with input and output conditions.

   Answer:

   Find the closest pair of Pts

   Input: P[1...n], a set of points in 2-dimensional space

   Output: P[a] and P[b], the closest pair of points, which means the distance between P[a] and P[b] is the shortest.

2. (3 points) The algorithm in the text divides the problem into halves around the median point according to x-coordinate. Recursion occurs on the left and right halves but we must still consider that the closest points cross the dividing line. Let δ_l be the distance between the closest points in the left partition and let δ_r be the distance between the closest points in the right partition. Then let $\delta = \min\{\delta_l, \delta_r\}$. Prove this theorem:

Theorem 1. *If $u = (x_u, y_u)$ and $v = (x_v, y_v)$ are the closest points such that u is on the left side of the partition and v is on the right side of the partition then both u and v are within δ of the partitioning line where δ is the minimum distance between points in the left half and points in the right half.*

Proof. Since point u and point v are the closest pointson on the left side and right side of partition respectively, and also we knew δ is the minimum distance between points in the left half and points in the right half, distance(u, v) must less than δ for such pair of points like u and v. If the distance of two points is equal to δ, that means the pair must be on the same side of partition or one of the point is on the line. □

3. (6 points) Now let S be the set of points that are within δ of the dividing line. Prove this theorem:

   Theorem 2. *If u is in S and on the left side of the dividing line then there are at most* 8 *other points in S on the right side of the line that are within δ of u.*

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.