

TCSS 343 - Assignment 2

Version 1.0

May 15, 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 handwriting is unclear.

Each problem is worth a total of 20 points except the challenge problem which is worth 10 points. Solutions receiving full 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 list of integers $A[a...b]$.

Output: $A[i]$ for some $a \leq i \leq b$ such that $A[i] \geq A[j]$ for all $a \leq j \leq b$.

Let $M(A[a...b])$ represent the output of the MAX problem on input $A[a...b]$. Let $\text{Max}(a, b)$ 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 .

- (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.

$$M(A[a...b]) = \begin{cases} A[a] & \text{if } a = b \\ \text{Max}(A[a], M(A[a+1...b])) & \text{if } a < b \end{cases}$$

Recursive algorithm:

findMax($A[a...b]$)

 If $a = b$, return $A[a]$.

 Let $c = \text{findMax}(A[a+1...b])$.

 Return $\text{Max}(A[a], c)$.

End findMax

- (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 $T(n)$ (you do not need to prove this bound).

$$T(n) = \begin{cases} c & \text{if } n \leq 1 \\ T(n-1) + d & \text{if } n > 1 \end{cases}$$

Tight bound:

an when $a = \min\{c, d\} \leq T(n) \leq b(n)$ when $b = \max\{c, d\}$.

- (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.

$$M(A[a...b]) = \begin{cases} -\infty & \text{if } a > b \\ A[a] & \text{if } a = b \\ \text{Max}(A[m], \text{Max}(M(A[a...m-1]), M(A[m+1...b]))) & \text{if } a < b \end{cases}$$

Recursive algorithm:

findMax($A[a \dots b]$)

 If $a > b$ return $-\infty$.

 If $a = b$ return $A[a]$.

 Let $m = \lfloor \frac{a+b}{2} \rfloor$.

 Let $c = \text{findMax}(A[m+1 \dots b])$.

 Let $d = \text{findMax}(A[a \dots m-1])$.

 return $\text{Max}(A[m], \text{Max}(c, d))$.

End findMax

- (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. Find a similar recurrence in your notes and state the tight bound on $T(n)$ (you do not need to prove this bound).

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

Tight Bound:

an when $a = \min\{c, \frac{d}{2}\} \leq T(b) \leq bn - a$ when $a = d, b = a + c$ for $n > 1$.

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

Sum of All Integers in a List (SUM)

Input: A list of integers $A[a \dots b]$.

Output: $s = \sum_{i=a}^b A[i]$.

Let $S(A[a \dots b])$ represent the output of the SUM problem on input $A[a \dots b]$.

- (4 points) 5. State two different self-reductions for the SUM problem. Use the self-reduction examples from lecture as a guide.

- a) First, we can take the first element of the list and add it to the sum of the remaining list elements recursively. We can perform the action above until there is only a single element in the list.

$$\text{Sum}(A[a \dots b]) = \begin{cases} A[a] & \text{if } a = b \\ A[a] + \text{Sum}(A[a+1 \dots b]) & \text{if } a < b \end{cases}$$

- b) Start with splitting the list by a half, we can use the sum method to add up all the elements in the first half of the list and the second half of the list until there are only two elements in the list. Then we add them up to get the ultimate sum.

$$Sum(A[a \dots b]) = \begin{cases} 0 & \text{if } a > b \\ A[a] & \text{if } a = b \\ A[m] + Sum(A[a \dots m-1]) + Sum(A[m+1 \dots b]) & \text{if } a < b \end{cases}$$

(4 points) 6. Give recursive algorithms based on your divide and conquer self-reductions to solve the SUM problem.

a) SUM($A[a \dots b]$)

if array A's length is 1

return A[0]

else

return A[0] + SUM($A[a + 1 \dots b]$)

b) SUM($A[a \dots b]$)

if array A's length is 2

return A[0] + A[1]

else

return SUM($A[a \dots \frac{A's \text{ length}}{2}]$) + SUM($A[\frac{A's \text{ length}}{2} \dots b]$)

(2 points) 7. What are the worst case runtimes of the solutions you have generated. (Just state the runtimes. You do not need to show your work.)

a) $\theta(n)$.

b) $\theta(n)$.

2.2 EXPLORE

Consider the recurrence $T(n)$.

$$T(n) = \begin{cases} c & \text{if } n = 0 \\ 4T(\lfloor \frac{n}{4} \rfloor) + 12 & \text{if } n > 0 \end{cases}$$

(3 points) 1. State and prove by induction a theorem showing $T(n) \in O(n)$.

Theorem 1. $T(n) \leq bn - a$ for all $n > n_0$.

Proof. (By induction)

Base case: ($n = 1, 2, 3, 4$). We need $T(4) \leq b - a$ to satisfy all the base cases.

Inductive hypothesis: Assume $T(K) \leq b * K - a$ for $K < n$,

Induction step: when $n \geq 5$,

$$\begin{aligned} T(n) &= 4T(\lfloor \frac{n}{4} \rfloor) + 12 \\ &\leq 4(b * (\frac{n}{4}) - a) + 12 \\ &= bn - 4a + 12 \\ &\leq bn - a \quad \text{only if } a \geq 4 \end{aligned}$$

Let $a = 4$ and $b = 4c + 16$, so $T(n) \leq bn - a$ for all $n \geq 1$ by induction. □

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

Theorem 2. $T(n) \geq dn$ for all $n > n_0$.

Proof. (By induction)

Base case: ($n = 0$). $T(0) \geq d * 0 \Rightarrow c \geq 0$.

Inductive hypothesis: Assume $T(K) \geq d * K$ for $K < n$,

Induction step: when $n > 0$,

$$\begin{aligned} T(n) &= 4T(\lfloor \frac{n}{4} \rfloor) + 12 \\ &\geq 4 * d(\frac{n}{4} - 1) + 12 \\ &= dn - 4d + 12 \\ &\geq dn \quad \text{only if } d \leq 3 \end{aligned}$$

Let $d = 3$, so $T(n) \geq dn$ for all $n > 0$ by induction. □

Consider the recurrence $T(n)$.

$$T(n) = \begin{cases} c & \text{if } n = 0 \\ 4T(\lfloor \frac{n}{4} \rfloor) + 12n & \text{if } n > 0 \end{cases}$$

(3 points) 3. State and prove by induction a theorem showing $T(n) \in O(n \log_2 n)$.

Theorem 3. $T(n) \leq bn \log(n) + a$ for all $n > n_0$.

Proof. (By induction)

Base case: $(n = 1, 2, 3, 4)$. $T(4) \leq b * 4 * \log(4) + a \Rightarrow a \geq T(4)$ for all the base cases.

Inductive hypothesis: Assume $T(K) \leq bK \log(K) + a$ for $K < n$,

Induction step: when $n > 4$,

$$\begin{aligned} T(n) &= 4T(\lfloor \frac{n}{4} \rfloor) + 12n \\ &\leq 4 * b * \frac{n}{4} \log(\frac{n}{4}) + 4a + 12n \\ &= bn * \log(\frac{n}{4}) + 4a + 12n \\ &= bn \log(n) - bn \log(4) + 4a + 12n \\ &\leq bn \log(n) + a \quad \text{only if } 2bn \geq 4a + 12n \end{aligned}$$

Let $a = 4c + 12$, $b = 6 + 8c + 24$, so $T(n) \leq bn \log(n) + a$ for all $n \geq 1$ by induction. □

(3 points) 4. State and prove by induction a theorem showing $T(n) \in \Omega(n \log_2 n)$.

Theorem 4. $T(n) \geq dn \log(n)$ for all $n > n_0$.

Proof. (By induction)

Base case: $(n = 1, 2, 3)$. $T(3) \geq d * 3 * \log(3) \Rightarrow c \geq 0$.

Inductive hypothesis: Assume $T(K) \geq dK \log(K)$ for $K < n$,

Induction step: when $n > 0$,

$$\begin{aligned} T(n) &= 4T(\lfloor \frac{n}{4} \rfloor) + 12n \\ &\geq 4 * d * (\frac{n}{4} - 1) \log(\frac{n}{4}) + 12n \\ &= (dn - 4d)(\log(n) - 3) + 12n \\ &= dn \log(n) - 3dn - 4d \log(n) + 12d + 12n \\ &\geq dn \log(n) \end{aligned}$$

The last step is true only if $3dn + 4d \log(n) - 12d \leq 3dn + 4dn \leq 7dn \leq 12n$. So we will get $d \leq \frac{12}{7}$. We can pick $d = \frac{12}{7}$. By induction we have shown for all $n > 0$ that $T(n) \geq dn \log(n)$ by induction. □

Consider the recurrence $T(n)$.

$$T(n) = \begin{cases} c & \text{if } n = 0 \\ 3T(\lfloor \frac{n}{4} \rfloor) + 2 & \text{if } n > 0 \end{cases}$$

- (4 points) 5. Use the recursion tree or repeated substitution method to come up with a good guess for a bound $g(n)$ on the recurrence $T(n)$. **You do not need to prove your guess correct.**

Answer:

$$\begin{aligned} T(n) &= 3T(\lfloor \frac{n}{4} \rfloor) + 2 \\ &= 3^2 T(\lfloor \frac{n}{4^2} \rfloor) + 3 * 2 + 2 \\ &= 3^3 T(\lfloor \frac{n}{4^3} \rfloor) + 3^2 * 2 + 3 * 2 + 2 \\ &= 3^4 T(\lfloor \frac{n}{4^4} \rfloor) + 3^3 * 2 + 3^2 * 2 + 3 * 2 + 2 \\ &= \dots \\ &= 3^k T(\lfloor \frac{n}{4^k} \rfloor) + \sum_{i=0}^{k-1} 3^i * 2 \end{aligned}$$

So we want $\frac{n}{4^k} < 1 \Rightarrow k > \log_4 n$. We can plug in $k = \log_4 n + 1$ we get:

$$3^{\log_4 n + 1} c + 2 \sum_{i=0}^{\log_4 n} 3^i = 3c * n^{\log_4 3} + 3 * n^{\log_4 3} - 1 \in \theta(n^{\log_4 3})$$

Consider the recurrence $T(n)$.

$$T(n) = \begin{cases} c & \text{if } n = 0 \\ 2T(\lfloor \frac{n}{4} \rfloor) + 2\sqrt{n} & \text{if } n > 0 \end{cases}$$

- (4 points) 6. Use the recursion tree or repeated substitution method to come up with a good guess for a bound $g(n)$ on the recurrence $T(n)$. **You do not need to prove your guess correct.**

Answer:

$$\begin{aligned} T(n) &= 2T(\lfloor \frac{n}{4} \rfloor) + 2\sqrt{n} \\ &= 2^2 T(\lfloor \frac{n}{4^2} \rfloor) + 2\sqrt{n} + 2\sqrt{n} \\ &= 2^3 T(\lfloor \frac{n}{4^3} \rfloor) + 2\sqrt{n} + 2\sqrt{n} + 2\sqrt{n} \\ &= 2^4 T(\lfloor \frac{n}{4^4} \rfloor) + 2\sqrt{n} + 2\sqrt{n} + 2\sqrt{n} + 2\sqrt{n} \\ &= \dots \\ &= 2^k T(\lfloor \frac{n}{4^k} \rfloor) + 2k\sqrt{n} \end{aligned}$$

So we will plug in the same $k = \log_4 n + 1$ we get:

$$2^{\log_4 n + 1} c + 2(\log_4 n + 1)\sqrt{n} = 2c\sqrt{n} + 2\log_4 n * \sqrt{n} + 2\sqrt{n} \in O(n^{\frac{1}{2}} \log n)$$

2.3 EXPAND

Consider the recurrence $T(n)$.

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

- (4 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 $T(n)$.

Answer:

$$\begin{aligned} T(n) &= 2T(\lfloor \frac{n}{4} \rfloor) + 16 \\ &= 2^2 T(\lfloor \frac{n}{4^2} \rfloor) + 2 * 16 + 16 \\ &= 2^3 T(\lfloor \frac{n}{4^3} \rfloor) + 2^2 * 16 + 2 * 16 + 16 \\ &= 2^4 T(\lfloor \frac{n}{4^4} \rfloor) + 2^3 * 16 + 2^2 * 16 + 2 * 16 + 16 \\ &\dots \\ &= 2^k T(\lfloor \frac{n}{4^k} \rfloor) + \sum_{i=0}^{k-1} 2^i * 16 \end{aligned}$$

So we will plug in the same $k = \log_4 n + 1$ we get:

$$2^{\log_4 n + 1} T(\lfloor \frac{n}{4^{\log_4 n + 1}} \rfloor) + \sum_{i=0}^{\log_4 n + 1 - 1} 2^i * 16 = 2c\sqrt{n} + 32\sqrt{n} - 16 \in O(n^{\frac{1}{2}})$$

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

Theorem 5. $T(n) \leq bn^{\frac{1}{2}} - a$ for all $n > n_0$.

Proof. (By induction)

Base case: ($n = 1, 2, 3, 4$). $T(4) \leq b * 1^{\frac{1}{2}} - a \Rightarrow T(4) \leq b - a$ to satisfy all the base cases.

Inductive hypothesis: Assume $T(K) \leq bK^{\frac{1}{2}} - a$ for $K < n$,

Induction step: when $n \geq 5$,

$$\begin{aligned} T(n) &= 2T(\lfloor \frac{n}{4} \rfloor) + 16 \\ &\leq 2(b * (\frac{n}{4})^{\frac{1}{2}} - a) + 16 \\ &= bn^{\frac{1}{2}} - 2a + 16 \\ &\leq bn^{\frac{1}{2}} \quad \text{only if } a \geq 8 \end{aligned}$$

Let $a = 8$, $b = 8 + c$, so $T(n) \leq bn^{\frac{1}{2}} - a$ for all $n \geq 1$ by induction. □

(3 points) 3. State and prove by induction a theorem showing $T(n) \in \Omega(g(n))$.

Theorem 6. $T(n) \geq dn^{\frac{1}{2}}$ for all $n > n_0$.

Proof. (By induction)

Base case: ($n = 1, 2, 3, 4$). $T(4) \geq d * 1^{\frac{1}{2}} \Rightarrow d \leq T(3)$ for all the base cases.

Inductive hypothesis: Assume $T(K) \geq dK^{\frac{1}{2}}$ for $K < n$,

Induction step: when $n > 4$,

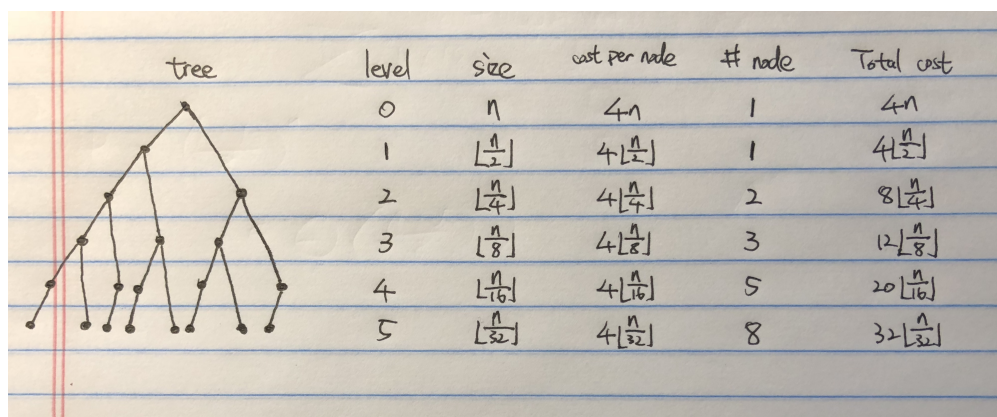
$$\begin{aligned} T(n) &= 2T(\lfloor \frac{n}{4} \rfloor) + 16 \\ &\geq 2d * ((\frac{n}{4})^{\frac{1}{2}} - 1) + 16 \\ &= dn^{\frac{1}{2}} - 2d + 16 \\ &\geq dn^{\frac{1}{2}} \quad \text{only if } d \leq 8 \end{aligned}$$

Let $d = \min\{8, T(3)\}$, so $T(n) \geq dn^{\frac{1}{2}}$ for all $n \geq 1$ by induction. □

Consider the 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 points) 4. 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 points) 5. Express the cost of all levels of the recursion tree as a sum over the cost of each level of the recursion tree.

Cost function: $F(k)c + \sum_{i=0}^{k-1} F(i) * \lfloor \frac{4n}{2^i} \rfloor$, where k is the level when size is 1.

(2 points) 6. Give a function $g(n)$ and show it is an upper bound on the sum.

Answer:

$$cF(k) + \sum_{i=0}^{k-1} F(i) * \lfloor \frac{4n}{2^i} \rfloor \leq cF(k) + 4n \sum_{i=0}^{k-1} F(i) * \frac{1}{2^i}$$

Since $\sum_{i=0}^{k-1} F(i) * \frac{1}{2^i} < 1 \Rightarrow 4n \sum_{i=0}^{k-1} F(i) * \frac{1}{2^i} < 4n$, and $c * F(k)$ is just a constant. It's suffice to say that $cF(k) + 4n \sum_{i=0}^{k-1} F(i) * \frac{1}{2^i} \in O(n) = g(n)$.

(3 points) 7. State and prove by induction a theorem showing $T(n) \in O(g(n))$.

Theorem 7. $T(n) \leq bn$.

Proof. (By induction) For base case: $n = 1, 2, 3$. We need $T(3) \leq b$ to prove all the base cases.

Inductive hypothesis: Assume $T(K) \leq bk$ for $K < n$.

Inductive step: when $n > 3$,

$$\begin{aligned} T(n) &= T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + 4n \\ &\leq b\left\lfloor \frac{n}{2} \right\rfloor + b\left\lfloor \frac{n}{4} \right\rfloor + 4n \\ &\leq b\frac{n}{2} + b\frac{n}{4} + 4n \\ &= \frac{3}{4}bn + 4n \\ &\leq bn \quad \text{only if } b \geq 16 \end{aligned}$$

Let $b = \max\{T(3), 16\}$, so $T(n) \leq bn$ for all $n \geq 1$ by induction. □

(3 points) 8. State and prove by induction (or other means) a theorem showing $T(n) \in \Omega(g(n))$.

Theorem 8. $T(n) \geq an$.

Proof. Since $T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + 4n > 4n$ for sure and $T(K)$ is always going to be positive for all $K > 0$. Hence, $T(n) \in \Omega(n)$. □

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

2.4 CHALLENGE

In lecture we considered a proof that the expected worst case running time of the *randomized quicksort algorithm* is $\Theta(n \log n)$. The analysis used an integral approximation for a summation that we have not studied in this class. There is a proof of this result that does not rely on this method.

The proof is based on the following observation. With probability $\frac{1}{2}$ the pivot selected will be between $\frac{n}{4}$ and $\frac{3n}{4}$ (i.e. a good pivot). Also with probability $\frac{1}{2}$ the pivot selected will be between 1 and $\frac{n}{4}$ or between $\frac{3n}{4}$ and n (i.e. a bad pivot).

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

We actually talked about this in class. This is the worst case for quicksort.

$$T(n) = \begin{cases} c & \text{if } n \leq 1 \\ T(n-1) + dn & \text{if } n > 1 \end{cases}$$

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

Because With probability $\frac{1}{2}$ the pivot selected will be between $\frac{n}{4}$ and $\frac{3n}{4}$, it's using the median of medians algorithm and the list M consists of $\frac{n}{4}$ medians of lists of size 4. By following the path we talked about in class, I got this recurrence:

$$T(n) = \begin{cases} c & \text{if } n \leq 1 \\ T(\lfloor \frac{3n}{4} \rfloor) + T(\lfloor \frac{n}{4} \rfloor) + dn & \text{if } n > 1 \end{cases}$$

- (2 points) 3. State a recurrence that expresses the expected worst case by combining the first two recurrences.

$$T(n) = \begin{cases} c & \text{if } n \leq 1 \\ \frac{1}{2}(T(\lfloor \frac{3n}{4} \rfloor) + T(\lfloor \frac{n}{4} \rfloor) + T(n-1)) + dn & \text{if } n > 1 \end{cases}$$

- (6 points) 4. Prove by induction that your recurrence is in $O(n \log n)$.

Proof. (By induction) For base case: $n = 2, 3, 4$. We need $T(4) \leq 2b \log 2 \Rightarrow b \geq \frac{T(4)}{2}$.

$n = 1$ is not a option here because $\log 1 = 0$.

Inductive Hypothesis: Assume $T(k) \leq bk \log k$ for $k < n$.

Inductive step: when $n \geq 5$,

$$\begin{aligned} T(n) &= \frac{1}{2}(T(\lfloor \frac{3n}{4} \rfloor) + T(\lfloor \frac{n}{4} \rfloor) + T(n-1)) + dn \\ &\leq \frac{1}{2}((b\lfloor \frac{3n}{4} \rfloor \log \lfloor \frac{3n}{4} \rfloor) + b\lfloor \frac{n}{4} \rfloor \log \lfloor \frac{n}{4} \rfloor + b(n-1) \log(n-1)) + dn \\ &\leq \frac{1}{2}((b\frac{3n}{4} \log \frac{3n}{4}) + b\frac{n}{4} \log \frac{n}{4} + b(n-1) \log(n-1)) + dn \\ &= \dots \end{aligned}$$

□

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.