1. Solve (that is, give a tight asymptotic bound for) each of the following recurrence relations. If it can be solved using the Master Theorem, you may do so. If not, you should guess the answer and then prove it by mathematical induction.

(a) Mergesort: $T(n) = 2T(n/2) + cn$

Proof. We will prove that Mergesort is $\Theta(n \log n)$ by the Master Theorem. We assume that $a = 2, b = 2$, and $f(n) = cn$. Then, $\log_b a$ is $\lg 2 = 1$. Since $f(n) = \Theta(n^{\log_b a}) = \Theta(n)$, then $T(n) = \Theta(n \log n)$. Hence, we have proved that Mergesort is $\Theta(n \log n)$ by the Master Theorem. ■

(b) Tree traversal: $T(n) = 2T(n/2) + c$

Proof. We will solve tree traversal by the Master Theorem. We assume that $a = 2, b = 2$, and $f(n) = c$. Then, $\log_b a$ is $\lg 2 = 1$. We can choose $\epsilon = 1$ such that $f(n) = O(1)$ and $O(n^{\log_b a - \epsilon}) = O(n^0) = O(1)$. Since $f(n) = O(n^{\log_b a - \epsilon})$, then $T(n) = \Theta(n^{\log_b a}) = \Theta(n)$. Hence, we have solved tree traversal by the Master Theorem. ■

(c) Binary search: $T(n) = T(n/2) + c$

Proof. We will prove that binary search is $\Theta(\log n)$ by the Master Theorem. We assume that $a = 1, b = 2, f(n) = c$, and $\log_b a = \lg 1 = 0$. We also note that $f(n) = \Theta(1)$ and $\Theta(n^{\log_b a}) = \Theta(n^0) = \Theta(1)$. Since $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n) = \Theta(\log n)$. Hence, we have proved that binary search is $\Theta(\log n)$ by the Master Theorem. ■

(d) Select: $T(n) = T(n/5) + T(7n/10) + cn$

Conjecture. Select is $\Theta(n)$.

Proof. We will first prove that Select is $O(n)$. That is, we will prove that $T(n) \leq cn$ by the Principle of Mathematical Induction.

We begin with the basis case, $n = 1$. We let $T(1) = c'$ such that $c' \leq c$. Hence, $T(1) \leq c$ and consequently, the basis case has been proven to be true.

We next prove the inductive step by strong induction. That is, we assume that $T(j) \leq cj$ for $j < k$ and prove that $T(k) \leq ck$. Then,

$$\begin{aligned} T(k) = T(k/5) + T(7k/10) + dk &\leq c(k/5) + c(7k/10) + dk \\ &\leq k(c/5 + 7c/10 + d) \\ &\leq k(9c/10 + d) \\ &\leq ck. \end{aligned}$$

Then, $T(k) \leq ck$, where $d \leq c/10$. Since d may be any finite constant, we have proven that Select is $O(n)$.

We will next prove that Select is $\Omega(n)$. That is, we will prove that $T(n) \geq cn$ by the Principle of Mathematical Induction.

We begin with the base case, $n = 1$. We let $T(1) = c'$ such that $c' \geq c$. Hence, $T(1) \geq c$ and consequently, the basis case has been proven to be true.

We next prove the inductive step by strong induction. That is, we assume that $T(j) \geq cj$ for $j < k$ and prove that $T(k) \geq ck$. Then,

$$\begin{aligned} T(k) = T(k/5) + T(7k/10) + dk &\geq c(k/5) + c(7k/10) + dk \\ &\geq k(c/5 + 7c/10 + d) \\ &\geq k(9c/10 + d) \\ &\geq ck. \end{aligned}$$

Then, $T(k) \geq ck$, where $d \geq c/10$. Since d may be any finite constant, we have proven that Select is $\Omega(n)$.

Since $T(n) = O(n)$ and $T(n) = \Omega(n)$, then $T(n) = \Theta(n)$ by definition of Θ-notation and consequently, the conjecture has been proven. ■

(e) Strassen: $T(n) = 7T(n/2) + cn^2$

Proof. We will prove that the Strassen algorithm is $\Theta(n^{\lg 7})$ by the Master Theorem. We assume that $a = 7, b = 2, f(n) = cn^2$, and $\log_b a = \lg 7$. If we let $\epsilon = 0.1$, then $f(n) = O(n^{\lg 7 - 0.1})$. Since $f(n) = O(n^{\log_b a - \epsilon})$, then $T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\lg 7})$. Hence, we have proved that the Strassen algorithm is $\Theta(n^{\lg 7})$ by the Master Theorem. ■

(f) Silly power: $T(n) = T(n-1) + c$

Conjecture. Silly power is $\Theta(n)$.

Proof. We will first prove that silly power is $O(n)$. That is, we will prove that $T(n) \leq cn$ by the Principle of Mathematical Induction.

We begin with the basis case, $n = 1$. We let $T(1) = c'$ such that $c' \leq c$. Hence, $T(1) \leq c$ and consequently, the basis case has been proven to be true.

We next prove the inductive step by strong induction. That is, we assume that $T(j) \leq cj$ for $j < k$ and prove that $T(k) \leq ck$. Then,

$$\begin{aligned} T(k) = T(k-1) + d &\leq c(k-1) + d \\ &\leq ck - c + d \\ &\leq ck. \end{aligned}$$

Then, $T(k) \leq ck$, where $-c + d \leq 0$ (or $d \leq c$). Since d may be any finite constant, we have proven that silly power is $O(n)$.

We next prove that silly power is $\Omega(n)$. That is, we will prove that $T(n) \geq cn$ by the Principle of Mathematical Induction.

We begin with the basis case, $n = 1$. We let $T(1) = c'$ such that $c' \geq c$. Hence, $T(1) \geq c$ and consequently, the basis case has been proven to be true.

We next prove the inductive step by strong induction. That is, we assume that $T(j) \geq cj$ for $j < k$ and prove that $T(k) \geq ck$. Then,

$$\begin{aligned} T(k) = T(k-1) + d &\geq c(k-1) + d \\ &\geq ck - c + d \\ &\geq ck. \end{aligned}$$

Then, $T(k) \leq ck$, where $-c + d \geq 0$ (or $d \geq c$). Since d may be any finite constant, we have proven that silly power is $\Omega(n)$.

Since $T(n) = O(n)$ and $T(n) = \Omega(n)$, then $T(n) = \Theta(n)$ by definition of Θ-notation and consequently, the conjecture has been proven by the Principle of Mathematical Induction. ■