CSC240 Winter 2024 Homework Assignment 7

due March 14, 2024

My name and student number: Haoyun (Bill) Xi 1009992019
The list of people with whom I discussed this homework assignment: Tianchu Li

Assume n is a power of 2. Consider the algorithm SRT that sorts an array $A[1...n]$ into nondecreasing order. Note that the algorithm SRT calls an auxiliary function AUX.

SRT$(A[1..n], n)$:
1 **if** $n > 1$ **then**
2   SRT$(A[1..\frac{n}{2}, n/2])$
3   SRT$(A[(\frac{n}{2}+1)..n, n/2])$
4   AUX$(A[1..n], n)$

AUX$(A[1..n], n)$:
1 **if** $n > 2$ **then**
2   **for** $i \leftarrow 1$ **to** $\frac{n}{4}$ **do**
3   swap the value of $A[i+\frac{n}{4}]$ and $A[i+\frac{n}{2}]$
4   AUX$(A[1..\frac{n}{2}], n/2)$
5   AUX$(A[(\frac{n}{2}+1)..n], n/2)$
6   AUX$(A[(\frac{n}{4}+1)..\frac{3n}{4}], n/2)$
7 **else if** $A[1] > A[2]$ **then**
8   swap the value of $A[1]$ and $A[2]$

1. Give recursive definitions of functions $U(n)$ and $L(n)$, which are, respectively, a good upper bound and a good lower bound on the worst case number of steps performed by AUX$(A[1..n], n)$, where steps are array element comparisons and swap operations.
   Justify your answers and remember to define the domains of your functions.

   Solution.
   Define for all $n \in \mathbb{Z}^+$ where $n = 2^i$ for some $i \in \mathbb{Z}^+, i > 1$

   $$U(n) = \begin{cases} 2, & n = 2 \\ 3U(\frac{n}{2}) + \frac{n}{4}, & n > 2 \end{cases} \quad L(n) = \begin{cases} 1, & n = 2 \\ 3L(\frac{n}{2}) + \frac{n}{4}, & n > 2 \end{cases}$$

   Let $\mathcal{P}_2 = \{n | n = 2^i \text{ for some } i \in \mathbb{N}^+\} \subseteq \mathbb{Z}^+$ denote the set of powers of 2.
   Alternatively, we can see $\mathcal{P}_2 \subseteq \mathbb{Z}^+$ as the recursively defined structure where the base case is $1 \in \mathcal{P}_2$ and the constructor case is $2 \cdot n \in \mathcal{P}_2$ for all $n \in \mathcal{P}_2$.
   $U : \mathcal{P}_2 \setminus \{1\} \to \mathbb{Z}^+, L : \mathcal{P}_2 \setminus \{1\} \to \mathbb{Z}^+$ are the domains and codomains of the above functions.
   Justification.
   If we take a close look at the program **AUX**, we can see that it is only called in **SRT** with argument $n > 1$. Thus, the base case will be $n = 2$. When $n > 2$, line 3 will be executed $n/4$ times under all cases. Line 4, 5, 6 will also be recursively called, where each line execute **AUX** with $n/2$, in total of 3 times, yielding in total of $3U(n/2)$ steps. Thus, the recursive

step in U, L are the same.
However, line 8 will only be executed when $A[1] > A[2]$. Since the effect of line 2 to 6 is not direct, we do not know under the worst case, how many times $A[1] > A[2]$ will return true to trigger line 8. However, we do know that line 8 will at most always trigger, leaving the upper bound function having two steps in the base case (line 7 and line 8). Line 8 will at least trigger 0 times, leaving the lower bound function having one step in the base case (line 7 only).

2. Solve the recurrence $U(n)$ using the method of repeated substitution and verify.
Do not use asymptotic notation.

Solution.
Let $n = 2^i$ for some $i \in \mathbb{Z}^+, i \geq 1$. Thus $\frac{n}{2^{i-1}} = 2$ and $i = \log_2 n$.

$$\begin{aligned}
U(n) &= 3U(\frac{n}{2}) + \frac{1}{4}n \\
&= 3(3U(\frac{n}{4}) + \frac{1}{4} \cdot \frac{n}{2}) + \frac{1}{4}n \\
&= 3^2 U(\frac{n}{2^2}) + \frac{1}{4}n(1 + \frac{3}{2}) \\
&= 3^3 U(\frac{n}{2^3}) + \frac{1}{4}n(1 + \frac{3}{2} + \frac{3^2}{2^2}) \\
&= \ldots\ldots \\
&= 3^{i-1} U(\frac{n}{2^{i-1}}) + \frac{1}{4}n \sum_{k=0}^{i-2} (\frac{3}{2})^k \\
&= 3^{\log_2 n - 1} \cdot 2 + \frac{1}{4}(\frac{1 - (\frac{3}{2})^{i-1}}{1 - \frac{3}{2}})n \\
&= \frac{2}{3} 3^{\log_2 n} + \frac{(\frac{3}{2})^{i-1} - 1}{2} n \\
&= \frac{2}{3} n^{\log_2 3} + \frac{3^{\log_2 n - 1} - 2^{\log_2 n - 1}}{2 \cdot 2^{\log_2 n - 1}} \\
&= \frac{2}{3} n^{\log_2 3} + (\frac{1}{3} n^{\log_2 3} - \frac{n}{2})/1 \\
&= n^{\log_2 3} - \frac{n}{2} \\
&= 3^{\log_2 n} - \frac{n}{2} \text{ or alternatively.}
\end{aligned}$$

We will verify by induction that the predicate

$$P(n) : \mathcal{P}_2 \setminus \{1\} \to \{T, F\} = \text{“}U(n) = 3^{\log_2 n} - \frac{n}{2}\text{”}$$

is true for all $n \in \mathcal{P}_2 \setminus \{1\}$.
Base Case: $n = 2$.
Since $3^{\log_2 n} - \frac{n}{2} = 3^{\log_2 2} - \frac{2}{2} = 3 - 1 = 2 = U(2), P(2)$ is clearly true.

Constructor Case:
Let $n \in \mathcal{P}_2 \setminus \{1\}$ be arbitrary. Assume $P(n)$ is true and we will show $P(2n)$.

$$\begin{aligned}
U(2n) &= 3U(n) + \frac{2n}{4} && \text{by recursive definition} \\
&= 3(3^{\log_2 n} - \frac{n}{2}) + \frac{n}{2} && \text{by induction hypothesis} \\
&= 3^{\log_2 n+1} - \frac{3n}{2} + \frac{n}{2} \\
&= 3^{\log_2 (2n)} - \frac{(2n)}{2} && \text{Since } \log_2 n + 1 = \log_2 n + \log_2 2 = \log_2 2n
\end{aligned}$$

Hence, $P(2n)$ is true as well.
Thus by structural induction, $\forall n \in \mathcal{P}_2 \setminus \{1\}.P(n)$.

Equivalently, we showed that

$$U(n) = 3^{\log_2 n} - \frac{n}{2}$$

is true for all n in the domain of U.

3. Give a recursive definition of the function $T(n)$, which represents a good upper bound on the worst case number of steps performed by SRT$(A[1..n], n)$, where steps are array element comparisons and swap operations.
   Justify your answer and remember to define the domain of your function.

   Solution.
   Define for all $n \in \mathcal{P}_2$

   $$T(n) = \begin{cases} 0, & n = 1 \\ 2T(\frac{n}{2}) + U(n), & n > 1 \end{cases}$$

   where we can substitute $U(n)$ as either $n^{\log_2 3} - \frac{n}{2}$ or $3^{\log_2 n} - \frac{n}{2}$.
   Justification.
   When $n = 1$, nothing will be executed. Hence it will have 0 steps. When $n > 1$, line 2 and 3 recursively calls **SRT** with argument $n/2$, in total of 2 times. Thus, resulting $2T(n/2)$ steps. In addition, line 4 calls **AXU** with argument n, resulting a maximum of $U(n)$ steps.
   Hence an upper bound of $T(n)$ in the recursive case is $2T(n/2) + U(n)$.

4. Solve the recurrence $T(n)$ using the method of repeated substitution and verify.
   Do not use asymptotic notation.

   Solution.

Let $n = 2^i$ for some $i \in \mathbb{Z}^+$. Thus $n/(2^i) = 1$ and $i = \log_2 n$.

$$
\begin{aligned}
T(n) &= 2T(\frac{n}{2}) + n^{\log_2 3} - \frac{n}{2} \\
&= 2(2T(\frac{n}{4}) + (\frac{n}{2})^{\log_2 3} - \frac{1}{2} \cdot \frac{n}{2}) + n^{\log_2 3} - \frac{1}{2}n \\
&= 2^2 T(\frac{n}{2^2}) + 2^1(\frac{n}{2})^{\log_2 3} + n^{\log_2 3} - 2 \cdot (\frac{1}{2}n) \\
&= 2^2(2T(\frac{n}{2^3}) + (\frac{n}{2^2})^{\log_2 3} - \frac{1}{2^2} \cdot \frac{n}{2}) + n^{\log_2 3} - 2 \cdot (\frac{1}{2}n) \\
&= 2^3 T(\frac{n}{2^3}) + 2^2(\frac{n}{2^2})^{\log_2 3} + 2^1(\frac{n}{2})^{\log_2 3} + n^{\log_2 3} - 3 \cdot (\frac{1}{2}n) \\
&= ...... \\
&= 2^i T(\frac{n}{2^i}) + \sum_{k=0}^{i-1} 2^k (\frac{n^{\log_2 3}}{2^{k \log_2 3}}) - \frac{i}{2}n
\end{aligned}
$$

Since $T(\frac{n}{2^i}) = T(1) = 0$, we have

$$
\begin{aligned}
T(n) &= \left(\sum_{k=0}^{i-1} 2^{k(1-\log_2 3)} \right) n^{\log_2 3} - \frac{i}{2}n \\
&= \left(\sum_{k=0}^{i-1} (\frac{2^1}{2^{\log_2 3}})^k \right) n^{\log_2 3} - \frac{i}{2}n \\
&= \left(\sum_{k=0}^{i-1} (\frac{2}{3})^k \right) n^{\log_2 3} - \frac{i}{2}n \\
&= \frac{1 - (\frac{2}{3})^i}{1 - \frac{2}{3}} \cdot n^{\log_2 3} - \frac{i}{2}n \\
&= 3(1 - \frac{2^{\log_2 n}}{3^{\log_2 n}}) \cdot 3^{\log_2 n} - \frac{i}{2}n \\
&= 3(3^{\log_2 n} - n) - \frac{i}{2}n \\
&= 3^{\log_2 n + 1} - 3n - \frac{\log_2 n}{2}n
\end{aligned}
$$

We will verify by induction that the predicate

$$Q(n) : \mathcal{P}_2 \to \{T, F\} = \text{“}T(n) = 3^{\log_2 n+1} - 3n - \frac{\log_2 n}{2}n\text{”}$$

is true for all $n \in \mathcal{P}_2$.
Base Case: $n = 1$.

$$3^{\log_2 n+1} - 3n - \frac{\log_2 n}{2}n = 3^{\log_2 1+1} - 3(1) - \frac{\log_2 1}{2}1 = 3 - 3 - 0 = 0 = T(1)$$

Hence, $Q(1)$ is true.
Constructor Case:

Let $n \in \mathcal{P}_2$ be arbitrary. Assume $Q(n)$ is true and we will show $Q(2n)$.

$$\begin{aligned}
T(2n) &= 2T(\frac{2n}{2}) + U(2n) && \text{by recursive definition} \\
&= 2T(n) + 3^{\log_2 2n} - \frac{2n}{2} && \text{by definition of } U \\
&= 2\left(3^{\log_2 n+1} - 3n - \frac{\log_2 n}{2}n\right) + 3^{\log_2 n+1} - n && \text{by induction hypothesis} \\
&= 3 \cdot 3^{\log_2 n+1} - 3(2n) - (\log_2 n)n - n && \text{combine like terms}
\end{aligned}$$

Notice that

$$1. \qquad 3 \cdot 3^{\log_2 n+1} = 3^{\log_2 n+1+1} = 3^{\log_2 n+\log_2 2+1} = 3^{\log_2 (2n)+1}$$

and

$$2. \qquad -(\log_2 n)n - n = -n(\log_2 n + 1) = -\frac{2n}{2} \cdot (\log_2 n + \log_2 2) = -(2n)(\log_2 (2n))/2$$

Thus substitute the above two equations, we have

$$T(2n) = 3^{\log_2 (2n)+1} - 3(2n) - \frac{\log_2 (2n)}{2}(2n)$$

Hence, $Q(n)$ IMPLIES $Q(2n)$.
Thus by the principle of structural induction, $\forall n \in \mathcal{P}_2 . Q(n)$.

Equivalently, we showed that

$$T(n) = 3^{\log_2 n+1} - 3n - \frac{\log_2 n}{2}n$$

is true for all n in the domain of T.