CSC236 Fall 2016

Office Hour 06

In the office hours this week, we discussed the following topics:

- We went over the closed form of *mergeSort* for $\hat{n}$. Recall that the recurrence relation we have for the algorithm is as follows:

$$T(n) = \begin{cases} 1 & n = 1 \\ T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n + 1 & n > 1 \end{cases}$$

Also, recall that $\hat{n} = 2^k$. Hence,

$$T(\hat{n}) = \begin{cases} 1 & n = 1 \\ 2T\left(\frac{\hat{n}}{2}\right) + \hat{n} + 1 & n > 1 \end{cases} \qquad (*)$$

We start from the recursive component of (*), and unwind it. We do not simplify it too much in order to see a pattern:

$$T(\hat{n}) = 2T\left(\frac{\hat{n}}{2}\right) + \hat{n} + 1$$

$$= 2\left(2\,T\left(\frac{\hat{n}}{2^2}\right) + \frac{\hat{n}}{2} + 1\right) + \hat{n} + 1$$ unwind it more by plug in $\frac{\hat{n}}{2}$ into $T(\hat{n})$ in (*)

$$= 2^2\,T\left(\frac{\hat{n}}{2^2}\right) + 2\frac{\hat{n}}{2} + 2 + \hat{n} + 1$$ make sure not to simplify it too much

$$= 2^2\left(2\,T\left(\frac{\hat{n}}{2^3}\right) + \frac{\hat{n}}{2^2} + 1\right) + 2\frac{\hat{n}}{2} + 2 + \hat{n} + 1$$ unwind it more by plug in $\frac{\hat{n}}{2^2}$ into $T(\hat{n})$ in (*)

$$= 2^3T\left(\frac{\hat{n}}{2^3}\right) + 2^2\frac{\hat{n}}{2^2} + 2^2 + 2\frac{\hat{n}}{2} + 2 + \hat{n} + 1$$ make sure not to simplify it too much

. . . continue unwinding to see a general pattern in terms of k

$$= 2^kT\left(\frac{\hat{n}}{2^k}\right) + 2^{k-1}\frac{\hat{n}}{2^{k-1}} + 2^{k-1} + \cdots + 2^2\frac{\hat{n}}{2^2} + 2^2 + 2\frac{\hat{n}}{2} + 2 + \hat{n} + 1$$

since $\hat{n} = 2^k$, $T\left(\frac{\hat{n}}{2^k}\right) = T(1) = 1$, now let's rearrange the terms

$$= 2^k + 2^{k-1} + \cdots + 2^2 + 2^1 + 2^0 + 2^{k-1}\frac{\hat{n}}{2^{k-1}} + \cdots + 2^2\frac{\hat{n}}{2^2} + 2^1\frac{\hat{n}}{2^1} + 2^0\frac{\hat{n}}{2^0}$$

$$= 2^{k+1} - 1 + k.\hat{n}$$

$$= 2.2^k - 1 + k.\hat{n}$$

$$= 2\hat{n} - 1 + (\lg\hat{n})\hat{n}$$ hence $T(\hat{n}) \in \theta(\hat{n}\lg\hat{n})$

So, we obtained the closed form for the time complexity of *mergeSort* for certain sizes (powers of k) as: $T(\hat{n}) = \hat{n}\,(\lg \hat{n}) + 2\hat{n} - 1$. Because of "…" in above calculations, one may ask how we know this is correct. So, we should prove it, that I left it as a practice for you.

From the closed form of $T(\hat{n})$, we concluded that $T(\hat{n}) \in \Theta(\hat{n} \lg \hat{n})$. (**)

We stated that in the course note, under Lemma 3.6, it has proved that for *mergeSort*:

$$T(\frac{\hat{n}}{2}) \le T(n) \le T(\hat{n}) \qquad \text{when } \frac{\hat{n}}{2} \le n \le \hat{n} \qquad (***)$$

From (***), we generalized (***) and made the *conjecture* that $T(n) \in \Theta(n \lg n)$. In the lecture, we proved $T(n) \in \Omega(n \lg n)$, and as a practice I asked you to prove $T(n) \in O(n \lg n)$.

- Another very important topic that we discussed during the office hours is as follows. Compare how we conducted the asymptotic analysis for *binSearch* last week with the analysis that we did for *mergeSort* this week (mentioned above too).
  - For *binSearch*,
    1. We estimated a *rough* closed form for $T(n)$.
    2. From the rough closed form, we made a conjecture that $T(n) \in \Theta(\lg n)$.
    3. Then, we proved our conjecture, using induction.
  - For *mergeSort*,
    1. We obtained an *exact* closed form for *certain* sizes in $T(n)$, denoted it by $T(\hat{n})$ where $\hat{n} = 2^k$. We left the correctness proof of $T(\hat{n})$ as a practice.
    2. From the exact closed form for $\hat{n}$ and the lemma that $T(n)$ is *monotonic* and *increasing, i.e.* $T(\frac{\hat{n}}{2}) \le T(n) \le T(\hat{n})$, we made a conjecture that $T(n) \in \Theta(n \lg n)$
    3. Then, we proved our conjecture.

Either of the two approaches above can be applied to the other algorithm (and may apply to any algorithms you see in future).

- We also briefly discussed Q3 of this week's tutorial. In particular, after the following observation

$$\begin{aligned}
T(n) &= \boldsymbol{T(n-1) + n - 2} \\
&= T(n-2) + n - 1 - 2 + n - 2 = \boldsymbol{T(n-2) + 2n - 5} \\
&= T(n-3) + n - 2 - 2 + 2n - 5 = \boldsymbol{T(n-3) + 3n - 9} \\
&= T(n-4) + n - 3 - 2 + 3n - 9 = \boldsymbol{T(n-4) + 4n - 14} \\
&= T(n-5) + n - 4 - 2 + 3n - 9 = \boldsymbol{T(n-5) + 5n - 20}
\end{aligned}$$

We made the following table, by which we tried to find a pattern for the last term:

k	value		
1	2	=0+2	=(1*2)/2-1+2
2	5	=2+3	=(2*3)/2-1+3
3	9	=5+4	=(3*4)/2-1+4
4	14	=9+5	=(4*5)/2-1+5
5	20	=14+6	=(5*6)/2-1+6
…	…	…	…

$$T(n) = T(n-k) + kn - \left(\left(\frac{k(k+1)}{2} - 1\right) + k + 1\right)$$

The rest should be obvious. We also discussed that too much simplification in unwinding could make it difficult to find the pattern.

- We also discussed common mistakes of Test 1 as follows:
    - Putting quantification inside definition of $P(n)$ in proof by induction.
    - Putting quantification inside inductive hypothesis in proof by induction.
    - Not assuming arbitrary elements in the inductive hypothesis of structural induction.