CS3230 Tutorial 2

AY 25/26 Sem 1 — github/omgeta

Q1). Give a tight asymptotic bound for $T(n) = 4 \cdot T(\frac{n}{4}) + \frac{n}{\log n}$

$$\begin{aligned}
T(n) &= 4T(\frac{n}{4}) + \frac{n}{\log n} \\
\implies \frac{T(n)}{n} &= \frac{T(n/4)}{n/4} + \frac{1}{\log n} \\
\frac{T(n/4)}{n/4} &= \frac{T(n/4^2)}{n/4^2} + \frac{1}{\log n/4} \\
&\vdots \\
\frac{T(n/4^{\log_4(n)-1})}{n/4^{\log_4(n)-1}} &= \frac{T(n/4^{\log_4(n)})}{n/4^{\log_4(n)}} + \frac{1}{\log n/4^{\log_4(n)-1}} && (\text{when } \tfrac{n}{4^k} = 1, k = \log_4 n)
\end{aligned}$$

Then by cancellation,

$$\begin{aligned}
\frac{T(n)}{n} &= \frac{T(1)}{T(1)} + \frac{1}{\log n} + \cdots + \frac{1}{\log n/4^{\log_4(n)-1}} \\
&= \frac{T(1)}{T(1)} + \frac{1}{\log 4^i} + \cdots + \frac{1}{\log 4^1} && (\text{let } n = 4^i) \\
&= \frac{T(1)}{T(1)} + \frac{1}{\log 4}\left\{\frac{1}{i} + \cdots + \frac{1}{1}\right\} && (\text{let } n = 4^i) \\
&= \frac{T(1)}{T(1)} + \Theta(\log i) && (\text{harmonic sum}) \\
&= \frac{T(1)}{T(1)} + \Theta(\log\log n) \\
\therefore T(n) &\in \Theta(n \log\log n)
\end{aligned}$$

Q2). $T(n) = 5T(\frac{n}{3}) + n$, $d = \log_3 5 = 1.46\ldots$ and $f(n) = n \in O(n^{\log_3 5 - \epsilon})$, so by case 1 $T(n) \in \Theta(n^{\log_3 5})$

Q3). $T(n) = 9T(\frac{n}{3}) + n^3$, $d = \log_3 9 = 2$ and $f(n) = n^3 \in \Omega(n^{2+\epsilon})$ ad for regularity $9(\frac{n}{3})^3 = \frac{1}{3}n^3 \leq \frac{1}{3}n^3 \wedge \frac{1}{3} < 1$, so by case 3 $T(n) \in \Theta(n^3)$

Q4). $T(n) = 16T(\frac{n}{4}) + n^2 \log n$, $d = \log_4 16 = 2$ and $f(n) = n^2 \log n \in \Theta(n^2 \log n)$, so by case 2 $T(n) \in \Theta(n^2 \log^2 n)$

Q5). Give a tight asymptotic bound for $T(n) = 4 \cdot T(\frac{n}{2}) + \sqrt{n}$.

1. Proof $T(n) \in O(n^2)$:
   - 1.1. Guess $T(n) \leq cn^2 - d\sqrt{n}$
   - 1.2. Base case: $T(0) = 0 \leq c \cdot 0^2 - d\sqrt{0}$
   - 1.3. Inductive step:
     $T(n) = 4T(\frac{n}{2}) + \sqrt{n} \leq 4(c\frac{n^2}{4} - d\frac{\sqrt{n}}{\sqrt{2}}) + \sqrt{n} = cn^2 - 2\sqrt{2}d\sqrt{n} + \sqrt{n} = cn^2 + (1 - 2\sqrt{2})d\sqrt{n}$,
     so choose $d < \frac{1}{1-2\sqrt{2}}$ and from base case $T(1) \leq q$, therefore
     $T(1) \leq c \times 1^2 - d \times 1 \leq q \implies c \leq q + d$
2. Proof $T(n) \in \Omega(n^2)$:
   - 2.1. Guess $T(n) \geq cn^2$
   - 2.2. Base case: $T(0) = 0 \geq c \cdot 0^2$
   - 2.3. Inductive step: $T(n) = 4T(\frac{n}{2}) + \sqrt{n} \geq 4 \cdot c(\frac{n}{2})^2 = cn^2$

Q6). $T(k, n) = T(\lceil \frac{k}{2} \rceil, n) + T(\lfloor \frac{k}{2} \rfloor, n) + kn = 2T(\frac{k}{2}, n) + \Theta(kn)$
By recursion tree, there are $\log k$ levels with $\Theta(kn)$ work at each level so $T(k, n) = \Theta(kn \log k)$