purdue university · CS 580
Introduction to the Analysis of Algorithms

HOMEWORK
Jun Cheng
January 26, 2016

Problem 1:

Using master theorem:

$$\begin{aligned} a &= 4 \\ b &= 4 \\ f(n) &= n \log n \end{aligned}$$

Then

$$\begin{aligned} \log_b^a &= 1 \\ f(n) &= \Omega(n^{\log_b^a - \epsilon}) \end{aligned}$$

Therefore,

$$T(n) = \Theta(n \log n)$$

Problem 2:

Bound for $T(n)$:

$$T(n) \leq an$$

where

$$\begin{aligned} a &= \frac{c}{1-\alpha-\beta} \\ \alpha &= \frac{3}{4} \\ \beta &= \frac{1}{5} \end{aligned}$$

Proof by induction:
Basis: $n < 7$

$$T(n) \leq c < \frac{c}{1-\alpha-\beta} \leq an$$

Induction step: (otherwise) Assume true for $n' < n$

$$\begin{aligned} T(n) &\leq cn + T(\lfloor \alpha n \rfloor) + 5T(\lfloor \beta n \rfloor) \\ &\leq cn + a\lfloor \alpha n \rfloor + a\lfloor \beta n \rfloor \\ &\leq cn + a\alpha n + a\beta n \\ &= cn + a(\alpha + \beta)n \\ &= cn + \frac{cn}{1-\alpha-\beta}(\alpha+\beta) \\ &= \frac{1}{1-\alpha-\beta}cn \\ &= an \end{aligned}$$

Problem 3:

For the SoSoSplotchy numbers, we know that :

$$\begin{aligned} S(0) &= 1 \\ S(1) &= 2 \\ S(n) &= 2S(n-1) + S(n-2) \end{aligned}$$

(a) To prove $S(n) = S(a+1)S(n-a-1) + S(a)S(n-2-a)$
Basis: $(a = 0)$ and $n \geq 2$

$$\begin{aligned} S(n) &= S(a+1)S(n-a-1) + S(a)S(n-2-a) \\ &= S(1)S(n-1) + S(0)S(n-2) \\ &= 2S(n-1) + S(n-2) \end{aligned}$$

Induction step: $(a > 0)$
If we know:

$$\begin{aligned} S(n) &= S(a+1)S(n-a-1) + S(a)S(n-a-2) \\ &= S(a+1)[2S(n-a-2) + S(n-a-3)] + S(a)S(n-2-a) \\ &= [2S(a+1) + S(a)]S(n-a-2) + S(a+1)S(n-a-3) \\ &= S(a+2)S(n-a-2) + S(a+1)S(n-a-3) \checkmark \end{aligned}$$

Then for $a+1$, we will have

$$\begin{aligned} S(n) &= S(a+2)S(n-a-2) + S(a+1)S(n-a-3) \\ &= S(a+2)S(n-a-2) + S(a+1)S(n-a-3) \end{aligned}$$

(b) Assuming the fact from part a, and let $n = 2k$ and $a = k-1$, we will have:

$$\begin{aligned} S(n) &= S(a+1)S(n-a-1) + S(a)S(n-2-a) \\ S(2k) &= S(k)S(k) + S(k-1)S(k-1) \end{aligned}$$

If we let $n = 2k+1$ and $a+1 = k+1$, then we will have:

$$\begin{aligned} S(2k+1) &= S(k+1)S(k) + S(k)S(k-1) \\ &= [2S(k) + S(k-1)]S(k) + S(k)S(k-1) \\ &= 2S(k)S(k) + 2S(k-1)S(k) \end{aligned}$$

(c) We know $S(2k+1) = 2S(k)S(k) + 2S(k-1)S(k)$
Replace k with $k-1$, then we will have:

$$\begin{aligned} S(2k-1) &= 2S(k-1)S(k-1) + 2S(k-1)S(k-2) \\ &= 2S(k-1)S(k-1) + 2S(k-1)[S(k) - 2S(k-1)] \\ &= 2S(k-1)[S(k) - S(k-1)] \end{aligned}$$

(d) If n is an odd number, let $2k+1=n$ and $k=\frac{n-1}{2}$, then

$$\begin{aligned}
S(n) &= S(2k+1)\\
&= 2S(k)S(k) + 2S(k-1)S(k)\\
&= 2S(\frac{n-1}{2})S(\frac{n-1}{2}) + 2S(\frac{n-3}{2})S(\frac{n-1}{2})\\
S(n-1) &= S(2k)\\
&= S(k)S(k) + S(k-1)S(k-1)\\
&= S(\frac{n-1}{2})S(\frac{n-1}{2}) + S(\frac{n-3}{2})S(\frac{n-3}{2})
\end{aligned}$$

If n is an even number, let $2k=n$ and $k=\frac{n}{2}$, then

$$\begin{aligned}
S(n) &= S(2k)\\
&= S(k)S(k) + S(k-1)S(k-1)\\
&= S(\frac{n}{2})S(\frac{n}{2}) + S(\frac{n-2}{2}S(\frac{n-2}{2})\\
S(n-1) &= S(2k-1)\\
&= 2S(k-1)S(k) - 2S(k-1)S(k-1)\\
&= 2S(\frac{n-2}{2})S(\frac{n}{2}) - 2S(\frac{n-2}{2})S(\frac{n-2}{2})
\end{aligned}$$

Pseudocode:

Function *SoSoSplotchy*(n):
if $n=1$ **then**
  return (2, 1)
else
  if $n = odd$ **then**

$$\begin{aligned}
&S(\frac{n-1}{2}), S(\frac{n-3}{2}) = SoSoplotchy(\frac{n-1}{2})\\
&S(n) = 2S(\frac{n-1}{2})S(\frac{n-1}{2}) + 2S(\frac{n-3}{2})S(\frac{n-1}{2})\\
&S(n-1) = S(\frac{n-1}{2})S(\frac{n-1}{2}) + S(\frac{n-3}{2})S(\frac{n-3}{2})
\end{aligned}$$

  if $n = even$ **then**

$$\begin{aligned}
&S(\frac{n}{2}), S(\frac{n-2}{2}) = SoSoplotchy(\frac{n}{2})\\
&S(n) = S(\frac{n}{2})S(\frac{n}{2}) + S(\frac{n-2}{2}S(\frac{n-2}{2})\\
&S(n-1) = 2S(\frac{n-2}{2})S(\frac{n}{2}) - 2S(\frac{n-2}{2})S(\frac{n-2}{2})
\end{aligned}$$

end
return $S(n), S(n-1)$

(e) Using uniform cost criterion:

$$T(n) = T(\lfloor\frac{n}{2}\rfloor) + c$$

Obvious method:

$$T(n) = 2T(n-1) + T(n-2)$$

Using logarithmic cost criterion:

$$T(n) = T(\lfloor \frac{n}{2} \rfloor) + c\left[\lg T(\lfloor \frac{n}{2} \rfloor)\right]^2$$

Obvious method:

$$T(n) = 2T(n-1) + T(n-2) + \lg T(\lfloor \frac{n}{2} \rfloor)$$