CSCI-GA.1170-001/002 Fundamental Algorithms September 19, 2019

Solutions to Problem 1 of Homework 2 (14 points)

Name: Jingshuai Jiang (jj2903) *Due: 5 pm on Thursday, September 19*

Collaborators: NetID1, NetID2

Consider the recurrence $T(n) = 8T(n/4) + n$ with initial condition $T(1) = 1$.

(a) (2 points) Solve it asymptotically using the "master theorem".

Solution: according to the master theorem, Because the $a = 8$,$b = 4$,so

$$f(n) = n < n^{\log_4 8}$$

so

$$T(n) = \Theta(n^{\log_4 8}) = \Theta(n^{\frac{3}{2}})$$

□

(b) (4 points) Solve it by the "guess-then-verify method". Namely, guess a function $g(n)$ — presumably solving part (a) will give you a good guess — and argue by induction that for all values of n we have $T(n) \leq g(n)$. What is the "smallest" $g(n)$ for which your inductive proof works?

Solution: guess that $T(n) \leq c \cdot n^2$ then $T(1) = 1 \leq c$,

then for $n \geq 2$

$$T(n) = 8T(\frac{n}{4}) + n \leq 8 \cdot c \cdot (\frac{n}{4})^2 + n = c \cdot \frac{n^2}{2} + n \leq c \cdot n^2$$

then we get

$$\frac{n^2}{2} \cdot c - n \geq 0$$

when $c = 1$ the equation is true for both T(1) and T(n) $n \geq 2$

then the "smallest" $g(n)$ should be n^2 □

(c) (4 points) Solve it by the "recursive tree method". Namely, draw the full recursive tree for this recurrence, and sum up all the value to get the final time estimate. Again, try to be as precise as you can (i.e., asymptotic answer is OK, but would be nice if you preserve a "leading constant" as well).

Solution:

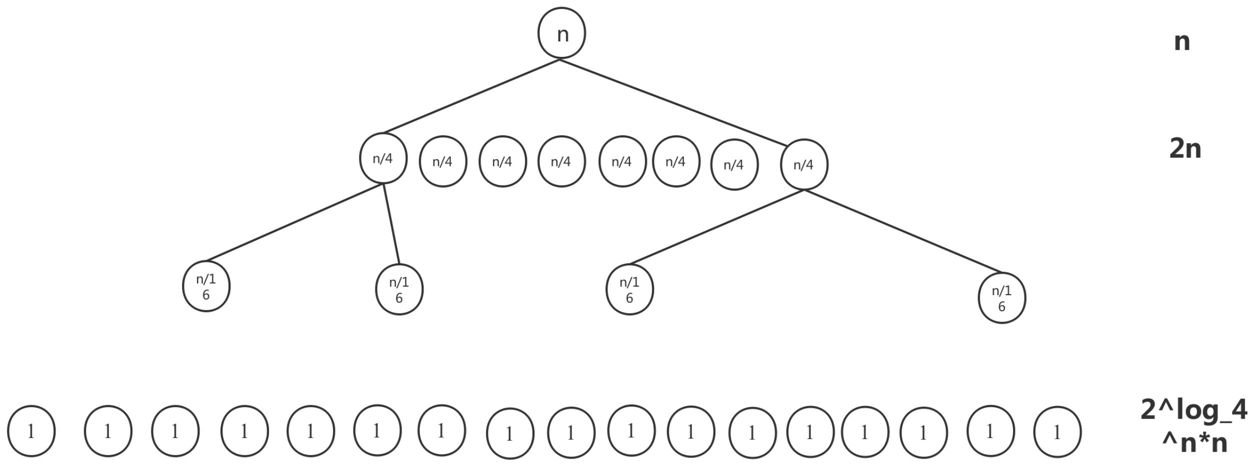

Figure 1: this is a recurrsive tree

$$(2^0 + 2^1 + 2^2 + \dots\dots + 2^{\log_4 n}) \cdot n = (2 \cdot \sqrt{n} - 1) \cdot n = 2 \cdot n^{\frac{3}{2}} - n = \Theta(n^{\frac{3}{2}})$$

$\square$

(d) (4 points) Solve it *precisely* using the "domain-range substitution" technique. Namely, make several changes of variables until you get a basic recurrence of the form $S(k) = S(k-1) + f(k)$ for some f, and then compute the answer from there. Make sure you carefully maintain the correct initial condition.

Solution:

let $n = 4^k$ then

$$T(n) = T(4^k) = 8 \cdot T(4^{k-1}) + 4^k$$

let

$$T(4^k) = S(k)$$

then

$$\frac{S(k)}{8^k} = \frac{S(k-1)}{8^{k-1}} + \frac{4^k}{8^k}$$

then let

$$\frac{S(k)}{8^k} = g(k)$$

$$g(k) = g(k-1) + \frac{1}{2}^k$$

then we get

$$g(k) = \frac{1}{2}^0 + ... + \frac{1}{2}^k = 2 - \frac{1}{2}^k$$

and we get

$$S(k) = 2 \cdot (4 \cdot 2)^k - 4^k$$

and we get

$$T(n) = 2 \cdot n\sqrt{n} - n = \Theta(n^{\frac{3}{2}})$$

$\square$

CSCI-GA.1170-001/002 Fundamental Algorithms	September 19, 2019

Solutions to Problem 2 of Homework 2 (10 points)

Name: Jingshuai Jiang (jj2903) *Due: 5 pm on Thursday, September 19*

Collaborators: NetID1, NetID2

Consider the following recursive procedure.

BLA(n):
  if $n = 1$ **then return** 1
  else return BLA($n-1$) + BLA($n-1$) + BLA($n-1$)

(a) (3 points) What function of n does BLA compute?

Solution:

$$BLA(n) = \begin{cases} 1 & \text{if } n = 1 \\ BLA(n-1) + BLA(n-1) + BLA(n-1) & \text{if } n >= 1 \end{cases} \tag{1}$$

according to the function of BLA(n) it equals to $3 \cdot BLA(n-1)$,and $3^2 \cdot BLA(n-2)$ by doing this recursively it equals to

$$3^{n-1}$$

□

(b) (4 points) What is the running time $T(n)$ of BLA?

Solution:

according to the expression of BLA, we get

$$T(n) = 3T(n-1)$$

by using the range-domain-substitution we get

$$g(n) = \frac{T(n)}{3^n} = \frac{T(n-1)}{3^{n-1}}$$

and we get

$$T(n) = 3^{n-1} = \Theta(3^n)$$

□

(c) (3 points) How do the answers to (a) and (b) change if we replace the last line by "**else return** $3 \cdot$ BLA($n-1$)"?

Solution: The answer to (a) will remain the same because $BLA(n) = 3^n$

but the answer to (b) will change to

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

because the algorithm will only have to caculate BLA(1) to BLA(n) for only one time of each, and the whole process will have n units to caculate.That makes the final time to $\Theta(n)$ $\square$

CSCI-GA.1170-001/002 Fundamental Algorithms	September 19, 2019

Solutions to Problem 3 of Homework 2 (17 Points)

Name: Jingshuai Jiang (jj2903) *Due: 5 pm on Thursday, September 19*

Collaborators: NetID1, NetID2

(a) (4 pts) Consider the following recurrence

$$T(n) = \sqrt{n}\, T(\sqrt{n}) + n.$$

Solve for $T(n)$ by domain-range substitution. (**Hint**: divide both parts by "something").

Solution:

$$\frac{T(n)}{n} = \frac{T(\sqrt{n})}{\sqrt{n}} + 1$$

then

$$S(n) = S(\sqrt{n}) + 1 = S(2) + 1 + ... + 1 = S(2) + loglogn = 1 + loglogn$$

then

$$T(n) = n + nloglogn = \Theta(nloglogn)$$

□

(b) (4 pts) Consider the following recurrence

$$T(n) = 2T(n/2) + n/\log n.$$

Solve for $T(n)$ by domain-range substitution.

Solution:

let

$$n = 2^k$$

then

$$T(2^k) = 2T(2^{k-1}) + \frac{2^k}{k}$$

then

$$\frac{T(2^k)}{2^k} = \frac{T(2^{k-1})}{2^{k-1}} + \frac{1}{k}$$

then we get(when k approaches infinite)

$$S(k) = S(k-1) + \frac{1}{k} = 1 + ... + \frac{1}{k} = lnk + c$$

then

$$T(2^k) = 2^k lnk + c \cdot 2^k$$

and

$$T(n) = n \cdot lnlogn + c \cdot n = \Theta(nlnlogn)$$

□

(c) (4 points) $T(n) = T(n/2) + \log n$.

Solution:

let

$$n = 2^k$$

then

$$T(2^k) = T(2^{k-1}) + k$$

let $T(2^k) = S(k)$ then

$$S(k) = S(k-1) + k$$

and we get

$$S(k) = 1 + ... + k = \frac{(1+k) \cdot k}{2}$$

then

$$T(n) = \frac{logn + log^2 n}{2} = \Theta(log^2 n) = \Theta((logn)^2)$$

Ps.the finally two are the same expression.And the first is widely used in China □

(d) (5 Points) $T(n) = 3T(n-1) - 2T(n-2), n > 2, T(2) = 1, T(1) = 0$.

Solution:

by moving one $T(n-1)$ to the left of the equation we get

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

let

$$T(n) - T(n-1) = S(n)$$

then we get

$$S(n) = 2 \cdot S(n-1)$$

because $T(2) = 1, T(1) = 0$ we can get that $S(2) = 1$ then

$$S(n) = 2^{n-2} \cdot S(2) = 2^{n-2}$$

so

$$T(n) - T(n-1) = 2^{n-2}$$

then

$$T(n) = 1 + ... + 2^{n-2} = 2^{n-1} - 1 = \Theta(2^n)$$

□