Data Structures and Algorithms

(ESO207)

Lecture 16:

Solving recurrences

that occur frequently in the analysis of algorithms.

Commonly occurring recurrences

$$T(n) = \frac{2n!}{5} \sqrt{2} \sqrt{2} (n/5)$$

Methods for solving Recurrences

commonly occurring in algorithm analysis

Methods for solving common Recurrences

- Unfolding the recurrence.
- Guessing the solution and then proving by induction.
- A General solution for a large class of recurrences (Master theorem)

Solving a recurrence by unfolding

```
Let T(1) = 1,
   T(n) = c n + 4 T(n/2) for n>1, where c is some positive constant
Solving the recurrence for T(n) by unfolding (expanding)
   T(n) = cn + 4 T(n/2)
        = cn + 2cn + 4^2 T(n/2^2)
        = cn + 2cn + 4cn + 4^3 T(n/2^3)
        = cn + 2 cn + 4cn + 8cn + ... + 4^{\log_2 n}
        = cn + 2 cn + 4 cn + 8 cn + ... + n^2
```

A geometric increasing series with $\log n$ terms and common ratio 2

$$= O(n^2)$$

Solving a recurrence by guessing and then proving by induction

$$T(1) = c_1$$

 $T(n) = 2T(n/2) + c_2 n$

Guess: $T(n) \le a n \log n + b$ for some con(

It looks similar/identical to the recurrence of merge sort. So we guess $T(n) = O(n \log n)$

Proof by induction:

Base case: holds true if $b \ge c_1$

Induction hypothesis: $T(k) \le a k \log k + b$ for all k < n

To prove: $T(n) \le a n \log n + b$

Proof:
$$T(n) = 2T(n/2) + c_2 n$$

 $\leq 2(a^{\frac{n}{2}}\log^{\frac{n}{2}}+b)+c_2 n$ // by induction hypothesis

$$= a n \log n - a n + 2 b + c_2 n$$

$$= a n \log n + b + (b + c_2 n - a n)$$

$$\leq a n \log n + b$$
 if $a \geq b + c_2$

These inequalities can be satisfied simultaneously by selecting $b = c_1$ and $a=c_1+c_2$

Hence $T(n) \le (c_1 + c_2) n \log n + c_1$ for all value of n. So $T(n) = O(n \log n)$

Solving a recurrence by guessing and then proving by induction

Key points:

- You have to make a <u>right guess</u> (past experience may help)
- What if your guess is too loose?
- Be <u>careful</u> in the **induction step**.

Solving a recurrence by guessing and then proving by induction

Exercise: Find error in the following reasoning.

For the recurrence $T(1) = c_1$, and $T(n) = 2T(n/2) + c_2 n$,

one guesses T(n) = O(n)

Proposed (wrong)proof by induction:

Induction hypothesis: $T(k) \le ak$ for all k < n $T(n) = 2T(n/2) + c_2 n$ $\le 2(a\frac{n}{2}) + c_2 n \text{ // by induction hypothesis}$ $= an + c_2 n$ = O(n)

A General Method for solving a large class of Recurrences

Solving a large class of recurrences

$$T(1) = 1,$$

$$T(n) = f(n) + a T(n/b)$$
Where

- a and b are constants and b > 1
- **f**(*n*) is a *multiplicative* function:

$$f(xy) = f(x)f(y)$$

AIM: To solve T(n) for $n = b^k$

Warm-up

```
f(n) is a multiplicative function:

f(xy) = f(x)f(y)

f(1) = 1

f(a^i) = f(a)^i

f(n^{-1}) = 1/f(n)
```

Example of a *multiplicative* function : $f(n) = n^{\alpha}$

Question: Can you express $a^{\log_b c}$ as power of c?

Answer: $c^{\log_b a}$

Solving a slightly general class of recurrences

$$T(n) = f(n) + a T(n/b)$$

$$= f(n) + a f(n/b) + a^{2}T(n/b^{2})$$

$$= f(n) + a f(n/b) + a^{2}f(n/b^{2}) + a^{3}T(n/b^{3})$$

$$= ...$$

$$= f(n) + a f(n/b) + ... + a^{i}f(n/b^{i}) + ... + a^{k-1}f(n/b^{k-1}) + a^{k}T(1)$$

$$= \left(\sum_{i=0}^{k-1} a^{i}f(n/b^{i})\right) + a^{k}$$
... after rearranging ...
$$= a^{k} + \sum_{i=0}^{k-1} a^{i}f(n/b^{i})$$
... continued to the next page ...

$$T(n) = a^{k} + \sum_{i=0}^{k-1} a^{i} f(n/b^{i})$$
(since f is multiplicative)
$$= a^{k} + \sum_{i=0}^{k-1} a^{i} f(n) / f(b^{i})$$

$$= a^{k} + \sum_{i=0}^{k-1} a^{i} f(n) / (f(b))^{i}$$

$$= a^{k} + f(n) \sum_{i=0}^{k-1} a^{i} / (f(b))^{i}$$

$$= a^{k} + f(n) \sum_{i=0}^{k-1} (a/f(b))^{i}$$
A geometric series
$$= a^{k} + (f(b))^{k} \sum_{i=0}^{k-1} (a/f(b))^{i}$$

Case 1:
$$a = f(b)$$
, $T(n) = a^k(k+1) = O(a^{\log_b n} \log_b n) = O(n^{\log_b a} \log_b n)$

$$T(n) = a^{k} + \sum_{i=0}^{k-1} a^{i} f(n/b^{i})$$
(since f is multiplicative)
$$= a^{k} + \sum_{i=0}^{k-1} a^{i} f(n) / f(b^{i})$$

$$= a^{k} + \sum_{i=0}^{k-1} a^{i} f(n) / (f(b))^{i}$$

$$= a^{k} + f(n) \sum_{i=0}^{k-1} a^{i} / (f(b))^{i}$$

$$= a^{k} + f(n) \sum_{i=0}^{k-1} (a/f(b))^{i}$$
For $a < f(b)$, the sum of this series is bounded by
$$\frac{1}{1 - \frac{a}{f(b)}} = o(1)$$
Case 2: $a < f(b)$, $T(n) = a^{k} + (f(b))^{k} \cdot o(1) = o((f(b))^{k}) = o(f(n))$

= O(f(n))

$$T(n) = a^{k} + \sum_{i=0}^{k-1} a^{i} f(n/b^{i})$$
(since f is multiplicative)
$$= a^{k} + \sum_{i=0}^{k-1} a^{i} f(n) / f(b^{i})$$

$$= a^{k} + \sum_{i=0}^{k-1} a^{i} f(n) / (f(b))^{i}$$

$$= a^{k} + f(n) \sum_{i=0}^{k-1} a^{i} / (f(b))^{i}$$

$$= a^{k} + f(n) \sum_{i=0}^{k-1} a^{i} / (f(b))^{i}$$

$$= a^{k} + f(n) \sum_{i=0}^{k-1} (a/f(b))^{i}$$

$$= a^{k} + (f(b))^{k} \sum_{i=0}^{k-1} (a/f(b))^{i}$$

$$= a^{k} + (f(b))^{k} \sum_{i=0}^{k-1} (a/f(b))^{i}$$

Case 3:
$$a > f(b)$$
, $T(n) = a^k + O(a^k) = O(n^{\log_b a})$

Three cases

```
T(n) = a^k + (f(b))^k \sum_{i=0}^{k-1} (a/f(b))^i
Case 1: a = f(b),
                         T(n) = O(n^{\log_b a} \log_b n)
Case 2: a < f(b),
                         \mathsf{T}(n) = \mathsf{O}(\mathsf{f}(n))
Case 3: a > f(b),
                         T(n) = O(n^{\log_b a})
```

```
T(1) = 1,

T(n) = f(n) + a T(n/b) where f is multiplicative.

There are the following solutions

Case 1: a = f(b), T(n) = n^{\log_b a} \log_b n

Case 2: a < f(b), T(n) = O(f(n))

Case 3: a > f(b), T(n) = O(n^{\log_b a})
```

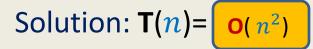
Examples

Case 1:
$$a = f(b)$$
, $T(n) = n^{\log_b a} \log_b n$

Case 2:
$$a < f(b)$$
, $T(n) = O(f(n))$

Case 3:
$$a > f(b)$$
, $T(n) = O(n^{\log_b a})$

Example 1:
$$T(n) = n + 4 T(n/2)$$





Case 1:
$$a = f(b)$$
, $T(n) = n^{\log_b a} \log_b n$

Case 2:
$$a < f(b)$$
, $T(n) = O(f(n))$

Case 3:
$$a > f(b)$$
, $T(n) = O(n^{\log_b a})$

Example 2:
$$T(n) = n^2 + 4 T(n/2)$$

Solution:
$$T(n) = O(n^2 \log_2 n)$$

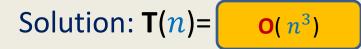


Case 1:
$$a = f(b)$$
, $T(n) = n^{\log_b a} \log_b n$

Case 2:
$$a < f(b)$$
, $T(n) = O(f(n))$

Case 3:
$$a > f(b)$$
, $T(n) = O(n^{\log_b a})$

Example 3:
$$T(n) = n^3 + 4 T(n/2)$$





```
If T(1) = 1, and T(n) = f(n) + a T(n/b) where f is multiplicative, then there are the following solutions
```

```
Case 1: a = f(b), T(n) = n^{\log_b a} \log_b n
```

Case 2:
$$a < f(b)$$
, $T(n) = O(f(n))$

Case 3:
$$a > f(b)$$
, $T(n) = O(n^{\log_b a})$

Example 4:
$$T(n) = 2 n^{1.5} + 3 T(n/2)$$

Solution:
$$T(n) =$$
 We can not apply master theorem directly since $f(n) = 2 n^{1.5}$ is not multiplicative.

Case 1:
$$a = f(b)$$
, $T(n) = n^{\log_b a} \log_b n$

Case 2:
$$a < f(b)$$
, $T(n) = O(f(n))$

Case 3:
$$a > f(b)$$
, $T(n) = O(n^{\log_b a})$

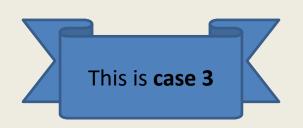
Example 4:
$$T(n) = 2 n^{1.5} + 3 T(n/2)$$

Solution:
$$G(n) = T(n)/2$$

$$\rightarrow$$
 G(n)= $n^{1.5}$ + 3 G(n/2)

$$\rightarrow$$
 G(n) = O($n^{\log_2 3}$) = O($n^{1.58}$)

→T(n) =
$$O(n^{1.58})$$
.



```
If T(1) = 1, and T(n) = f(n) + a T(n/b) where f is multiplicative, then there are the following solutions
```

Case 1:
$$a = f(b)$$
, $T(n) = n^{\log_b a} \log_b n$

Case 2:
$$a < f(b)$$
, $T(n) = O(f(n))$

Case 3:
$$a > f(b)$$
, $T(n) = O(n^{\log_b a})$

Example 6:
$$T(n) = T(\sqrt{n}) + c n$$

Solution:
$$T(n)$$
=

We can not apply master theorem directly since $T(\sqrt{n}) \ll T(n/b)$ for any constant b.

Solving $T(n) = T(\sqrt{n}) + c n$ using the method of unfolding

$$T(n) = c n + T(\sqrt{n})$$

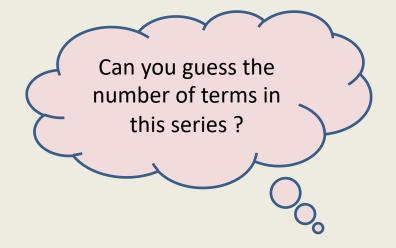
$$= c n + c\sqrt{n} + T(\sqrt[4]{n})$$

$$= c n + c\sqrt{n} + c \sqrt[4]{n} + T(\sqrt[8]{n})$$

$$= c n + c\sqrt{n} + ... c \sqrt[i]{n} + ... + T(1)$$

A series which is decreasing at a rate faster than any geometric series

$$= O(n)$$





```
If T(1) = 1, and T(n) = f(n) + a T(n/b) where f is multiplicative,
then there are the following solutions
Case 1: a = f(b), T(n) = n^{\log_b a} \log_b n
Case 2: \alpha < f(b), T(n) = O(f(n))
Case 3: a > f(b), T(n) = O(n^{\log_b a})
Example 5: T(n) = n (\log n)^2 + 2 T(n/2)
Solution: T(n)= Using the method of "unfolding", it can be shown that T(n) = O(n (\log n)^3).
```

Homework

Solve the following recurrences <u>systematically</u> (if possible by various methods). Assume that T(1) = 1 for all these recurrences.

- T(n) = 1 + 2 T(n/2)
- $T(n) = n^3 + 2 T(n/2)$
- $T(n) = n^2 + 7 T(n/3)$
- $T(n) = n/\log n + 2T(n/2)$
- T(n) = 1 + T(n/5)
- $T(n) = \sqrt{n} + 2 T(n/4)$
- $T(n) = 1 + T(\sqrt{n})$
- T(n) = n + T(9 n/10)
- $T(n) = \log n + T(n/4)$