ILLINOIS INSTITUTE OF TECHNOLOGY

Transforming Lives.Inventing the Future.www.iit.edu

CS430

Introduction to Algorithms

Lec #4

Lan Yao

Outlines

- **Key to desmos questions**

  https://student.desmos.com/?prepopulateCode=b33pqa
- **Average case analysis**
- **Three methods for solutions to asymptotic bounds**
- **Selection Sort and Bubble Sort**
- **Extended form of Master Theorem**

- Consider average cases
- Complexity of average cases?
  - average?
  - probability?
  - times that each input comes up?
  - expectation.

Pre-lecure
Expected Values

def: The expected value of a random variable X on a probability space (S, p) is the sum

$$E(X) = \sum_{s \in S} X(s)p(s)$$

- The term “expected value” is widely used, but misleading since the expected value might be totally unexpected or impossible!

- Probability distribution
  - uniform distribution-random inputs
  - equally likely
  - weighted
  - function

Expectation examples

The expected outcome of a fair die is:

1*(1/6)+2*(1/6)+3*(1/6)+4*(1/6)+5*(1/6)+6*(1/6)

=21/6 = 7/2

The expected outcome of the standard loaded die is:

1*(1/21)+2*(2/21)+3*(3/21)+4*(4/21)+5*(5/21)+6*(6/21)

=91/21 = 13/3

Linearity of Expectation

Theorem: Let X1 and X2 be random variables on a probability space (S, p).

Then E(X1 + X2) = E(X1) + E(X2)

Example:When two fair dice are rolled, here are both calculations:

$$E(X_1) + E(X_2) = \frac{7}{2} + \frac{7}{2} = 7 \quad \text{and}$$

$$E(X_1 + X_2) = \frac{1}{36} \sum_{j=1}^{6} \sum_{k=1}^{6} (j + k) = \frac{252}{36} = 7$$

Average Case Computational Complexity

Compute the expected value of the random variable that counts how many operations are executed by the algorithm.

Examples:

- Insertion Sort

Average Case Computational Complexity - Insertion Sort

- *n* distinct elements in list
- Sort using insertion sort
- X_i is the random variable equal to the number of comparisons used to insert a_i into the proper position after the first *i-1* elements have been sorted. $1 <= X_i <= i-1$

$E(X) = E(X_2)+E(X_3)+ . . . +E(X_n)$

$E(X_i)$ is expected number of comparisons to insert a_i into the proper position after the first *i-1* elements have been sorted.

Average Case Computational Complexity - Insertion Sort (cont.)

$E(X_i)$ = p(1 comp.)(1 comp.) + p(2 comp.)(2 comp.) +… + p(i-1 comp.)(i-1comp.)

p(k comp.) = 1/(i-1) (because if random data, it is equally likely the *i*th element could go in any sorted position from 1 to i-1)

$E(X_i) = [1/(i-1)](1) + [1/(i-1)](2) + \ldots + [1/(i-1)](i-1)$

$= [1/(i-1)][1+2+. . .+(i-1)]=[(i-1)(i)]/[2(i-1)]$

$= i/2$

$E(X) = E(X_2)+E(X_3)+ . . . +E(X_n)$

$E(X) =[1/2] \sum_{i=2}^{n} i =[1/2]\{(n+2)(n-1)/2\}$

$=(n^2+n-2)/4=\Theta(n^2)$

How to find a asymptotic bound for a function f(n)?

Complexity analysis with the definition to asymptotic bound--closed form solution for T(n) when algorithm is a recurrence relation.

- Not all problems can be solved with the divide and conquer approach. Maybe sub-problems are not independent, or solutions to sub-problems cannot be combined to find solution to main problem

Merge sort: $T(n)=2T(n/2)+\Theta(n)$, when $n>1$

Three methods to solve recurrences:

- Substitution
  - The substitution method for solving recurrences comprises two steps:

    1.Guess the form of the solution

    2.Use mathematical induction to show the solution works

    3.Mathematical induction states a theorem is true for any value n>=c, if the following conditions are true:

      ①Base case: the theorem holds for n=c
      ②Induction step: if the theorem holds for n-1, then it holds for n.
- Recursion Tree
- Master method

Inductive Proof*
(we will use to prove solution to a recurrence relation)

Ex1: show that $$\sum_{k=1}^{m+1} k = \frac{(m+1)(m+2)}{2}$$

proof:

1. Base case: when m=1, $\sum_{k=1}^{m+1} k = 1+2=3,\quad \frac{(m+1)(m+2)}{2} = \frac{2*3}{2} = 3$ true

2. Assume that when m, it is true.

$$\sum_{k=1}^{m+1} k = \frac{(m+1)(m+2)}{2}$$

3. When m+1, is it true?

$$\sum_{k=1}^{m+1+1} k = \sum_{k=1}^{m+1} k + m + 2 = \frac{(m+1)(m+2)}{2} + m + 2 = \frac{m^2+3m+2+2m+4}{2}$$

$$= \frac{m^2+5m+6}{2} = \frac{(m+2)(m+3)}{2} = \frac{[(m+1)+1][(m+1)+2]}{2} \quad true$$

Examples for Mathematical Induction

Ex2. Prove: 3^n-1 is a multiple of 2 when n>=0.

proof: when n=1, $3^1-1=2$-- ture

hypothesis:

when n=k, 3^k-1 is a multiple of 2, then

when n=k+1,

$3^{k+1}-1=3*3^k-1=3*(3^k-1)+2$

because 3^k-1 is a multiple of 2, which is:

$3^k-1=2m$, plug it in then

$3^{k+1}-1=3*2m+2=2\underline{(3m+1)}$.

QED.

By Lan Yao

Ex3. (Recurrence Relation: we do not know the exact function of T(n) other than the relation of T(n) and T(n-1))

$T(n)=T(n-1)+1$, where $T(0)=1$

has closed form solution as $T(n)=n+1$.

proof:

When $n=1$, $T(1)=T(0)+1=1+1$

Suppose, when $n=k$, $T(k)=k+1$, then when $n=k+1$

$T(k+1)=T(k+1-1)+1=T(k)+1$, plug the above hypothesis in,

$T(k+1)=k+1+1 =(k+1)+1$

QED.

Ex4. Evaluate the upper bound of $T(n)=2T(\lfloor \frac{n}{2} \rfloor)+n$

This is a recurrence relation too

Substitution Application

1. Guess $T(n)=O(n\lg n)$
2. Prove: $T(n)<=cn\lg n$

   suppose $T(\lfloor \frac{n}{2} \rfloor)<=c\lfloor \frac{n}{2} \rfloor \lg \lfloor \frac{n}{2} \rfloor$, then

   $2T(\lfloor \frac{n}{2} \rfloor)+n<=2c\lfloor \frac{n}{2} \rfloor \lg \lfloor \frac{n}{2} \rfloor +n<cn(\lg \lfloor \frac{n}{2} \rfloor +1/c)$

   $<=cn(\lg \lfloor \frac{n}{2} \rfloor +1)<cn\lg n$, which is

   $T(n)<cn\lg n$

BUT, we missed something--Base Case

Please discuss the base case on your own

Ex5. (recurrence relation)

Prove that T(n)=O(n), while $$T(n)=T(\left\lfloor \frac{n}{2} \right\rfloor)+T(\left\lceil \frac{n}{2} \right\rceil)+1$$

proof:

our goal is to prove:T(n)<=cn

Hypothesis: $T(\left\lfloor \frac{n}{2} \right\rfloor)<=c\left\lfloor \frac{n}{2} \right\rfloor$

$T(\left\lceil \frac{n}{2} \right\rceil)<=c\left\lceil \frac{n}{2} \right\rceil$

then T(n)<= $c\left\lfloor \frac{n}{2} \right\rfloor+c\left\lceil \frac{n}{2} \right\rceil+1=cn+1$ ✗

Adjust our goal to T(n)<=cn-d

Hypothesis: $T(\left\lfloor \frac{n}{2} \right\rfloor)<=c\left\lfloor \frac{n}{2} \right\rfloor-d$

$T(\left\lceil \frac{n}{2} \right\rceil)<=c\left\lceil \frac{n}{2} \right\rceil-d$

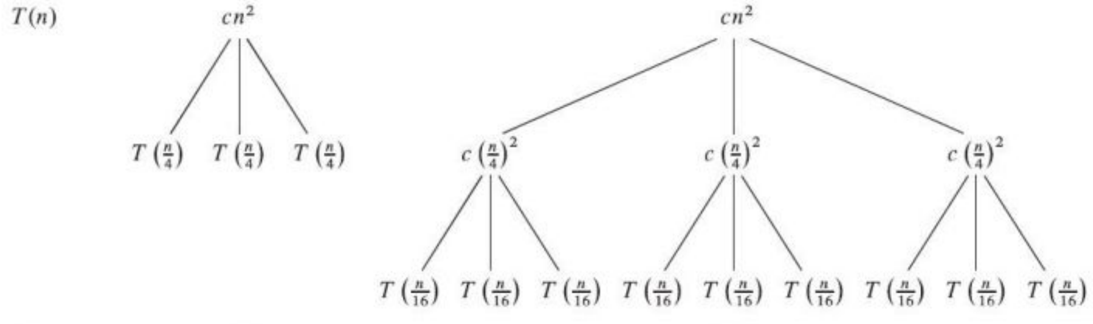
then:

$$T(n)<=c\left\lfloor \frac{n}{2} \right\rfloor-d+c\left\lceil \frac{n}{2} \right\rceil-d+1=cn-2d+1=cn-(2d-1)$$

when 2d-1>d, T(n)<=cn-d.
(2d-1>=d=d>=1, which is easy to satisfy) QED.

Recursion Tree

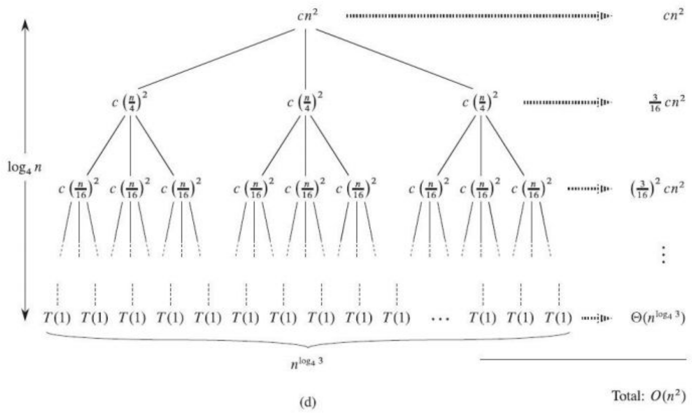
Ex1: Solve the upper bound of T(n), while $T(n)=3T(n/4)+cn^2$



$T(n)$
cn^2
$T\left(\frac{n}{4}\right)$
$c\left(\frac{n}{4}\right)^2$
$T\left(\frac{n}{16}\right)$

(a) (b) (c)

By Lan Yao



- Number of levels L: L=$\log_4 n$, for more accuracy, L=$\log_4 n+1$
- Cost of the ith level: $(3/16)^{i-1}cn^2$
- How many leaves? $3^{\log_4 n} = n^{\log_4 3}$ Then the cost of all leaves is to: $cn^{\log_4 3} = \Theta(n^{\log_4 3})$

Sum up all costs from all levels:

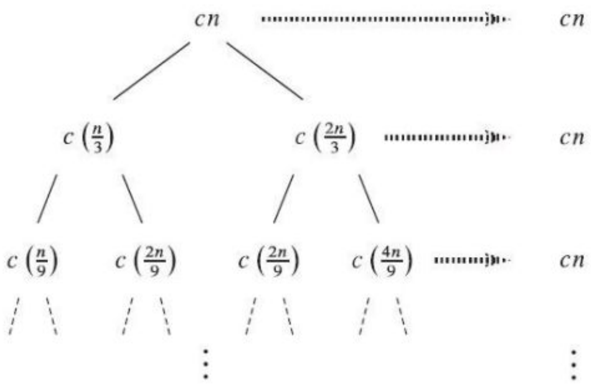
$$T(n) = cn^2[1+\frac{3}{16}+(\frac{3}{16})^2+\ldots+(\frac{3}{16})^{\log_4 n}]+\Theta(n^{\log_4 3}) < cn^2[1+\frac{3}{16}+(\frac{3}{16})^2+\ldots+(\frac{3}{16})^{\infty}]+\Theta(n^{\log_4 3})$$

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

$$T(n) = cn^2(\frac{1}{1-\frac{3}{16}})+\Theta(n^{\log_4 3}) = \frac{16}{13}cn^2+\Theta(n^{\log_4 3}) = \mathrm{O}(n^2)$$

EX2: Solve the upper bound of

$$T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + \mathrm{O}(n)$$



cn
cn
$c\left(\frac{n}{3}\right)$
$c\left(\frac{2n}{3}\right)$
cn
$c\left(\frac{n}{9}\right)$
$c\left(\frac{2n}{9}\right)$
$c\left(\frac{2n}{9}\right)$
$c\left(\frac{4n}{9}\right)$
cn

The number of levels--k: $(\frac{2}{3})^k n = 1$,

then

$$k = \log_{\frac{3}{2}} n = \frac{\lg n}{\lg \frac{3}{2}} = \frac{\lg n}{\lg 3 - 1} = \frac{1}{\lg 3 - 1} \lg n$$

$$T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + O(n) < cn \frac{1}{\lg 3 - 1} \lg n + cn = \frac{c}{\lg 3 - 1} n \lg n + cn$$

$$T(n) = O(n \lg n)$$

By Lan Yao

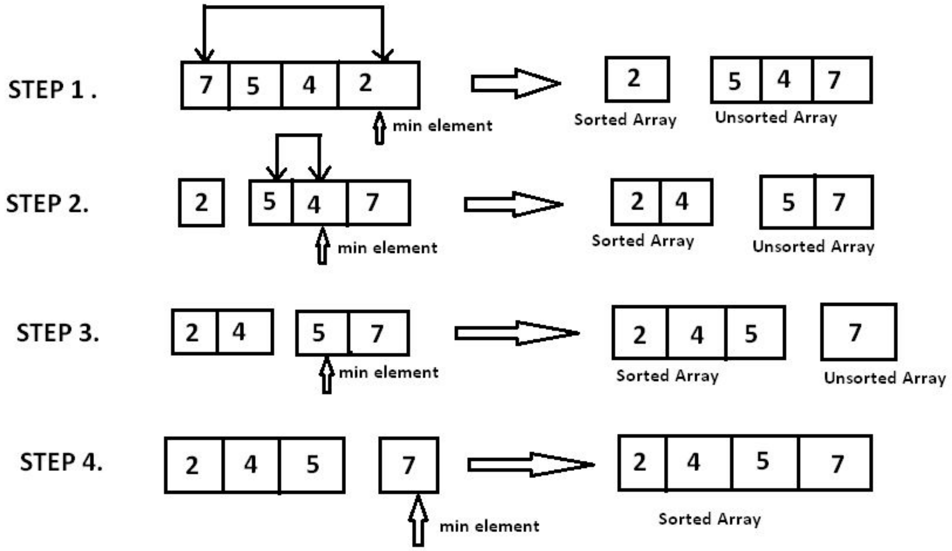
Algorithmic Analysis of other Sorts

- Selection Sort
- Bubble Sort

Selection Sort

Re| ... ; it
at t
giv

In e ... ed
sub



Complexity of Selection Sort

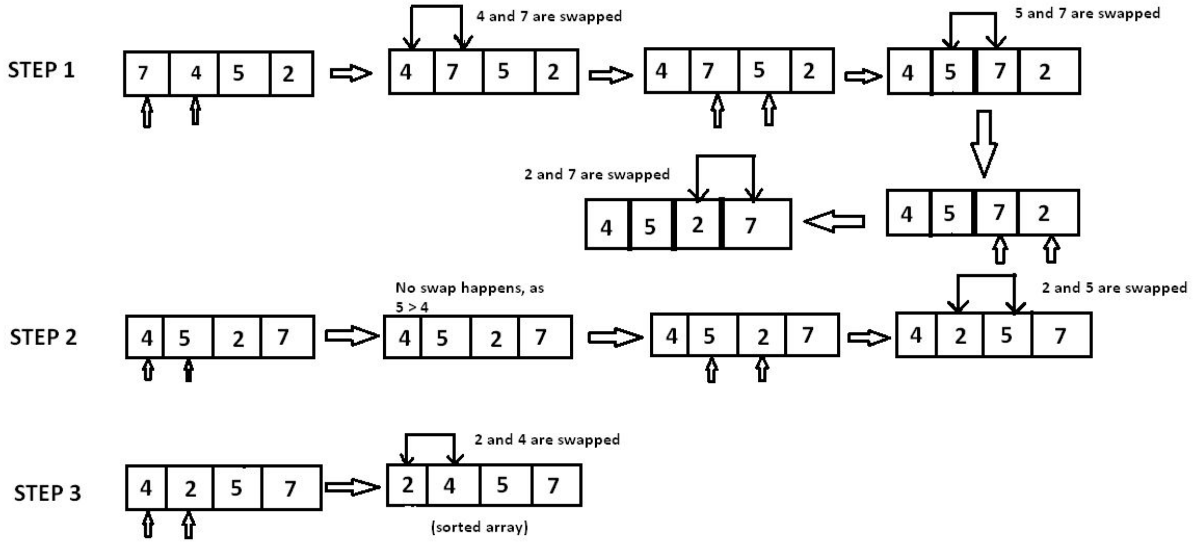
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = O(n^2)$$

- Best case: O(n^2)
- Worst case:O(n^2)
- Average case: O(n^2)

Bubble Sort

Repeatedly stepping through the array, comparing adjacent elements and swapping them if they are in a wrong order until the list is sorted, which is confirmed by no swap.

Bubble Sort


STEP 1
7 4 5 2
4 and 7 are swapped
4 7 5 2
4 7 5 2
5 and 7 are swapped
4 5 7 2
4 5 7 2
2 and 7 are swapped
4 5 2 7
STEP 2
4 5 2 7
No swap happens, as 5 > 4
4 5 2 7
4 5 2 7
2 and 5 are swapped
4 2 5 7
STEP 3
4 2 5 7
2 and 4 are swapped
2 4 5 7
(sorted array)

Complexity of Bubble Sort

$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = O(n^2)$$

- Best case: $T(n)=n-1=O(n)$
- Worst case: $T(n)=O(n^2)$
- Average case:

$$T(n) = \sum_{i=1}^{n-1} X_i p_i = \frac{1}{2}\sum_{i=1}^{n-1} X_i = \frac{1}{2}\times\frac{n(n-1)}{2} = \frac{1}{4}n^2 - \frac{1}{4}n$$

$$= O(n^2)$$

Master Theorem

Applicable for recurrence relation: $T(n)=aT(n/b)+f(n)$

- If your $T(n)$ satisfies any of the following cases, the asymptotic bounds can be solved according to the Master Theorem;
- Not all cases are included in cases of Master Theorem.

Three Cases of Master Theory

Case1:

If $f(n) = O(n^{\log_b a-\varepsilon})$ for some constants $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.

Case2:

If $f(n) = \Theta(n^{\log_b a})$ then $T(n) = \Theta(n^{\log_b a} \lg n)$.

Case 3:

If $f(n) = \Omega(n^{\log_b a+\varepsilon})$ for some constant $\varepsilon > 0$ and if af(n/b)<=cf(n) for some constant c<1 and sufficiently lager n then $T(n) = \Theta(f(n))$.

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

By Lan Yao

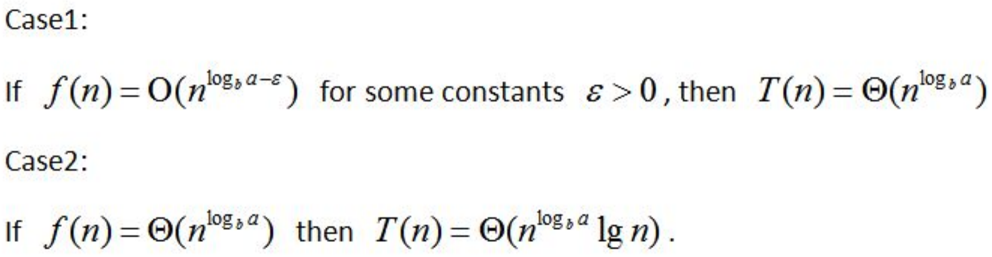
Ex: Solve the upper bound of $T(n)=8T(n/2)+n^2$

Derive Master Theorem, then we have

$a=8$, $b=2$, $\log_2 8=3$, $f(n)=O(n^2)=O(n^{3-1})$.

That is, when $\varepsilon=1$, it holds for case 1.

So, $T(n)=O(n^3)$



Case1:

If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constants $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$

Case2:

If $f(n) = \Theta(n^{\log_b a})$ then $T(n) = \Theta(n^{\log_b a} \lg n)$.

Ex: Solve T(n)'s asymptotic bound when

$$T(n) = T(\frac{2n}{3}) + 1$$

Solution: a=1, b=3/2, f(n)=1

then $\log_b a = \log_{\frac{3}{2}} 1 = 0$ and f(n)=1=n^0

It matches case 2 and gives us: T(n)=Θ(lgn)

By Lan Yao

Three Cases of Master Theory

Case1:

If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constants $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.

Case2:

If $f(n) = \Theta(n^{\log_b a})$ then $T(n) = \Theta(n^{\log_b a} \lg n)$.

Case 3:

If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$ and if af(n/b)<=cf(n) for some constant c<1 and sufficiently lager n then $T(n) = \Theta(f(n))$.

T(n)=aT(n/b)+f(n)

By Lan Yao

Ex: Solve T(n)’s asymptotic bound when

$$T(n) = 3T(\frac{n}{4}) + n\lg n$$

Solution: a=3, b=4, f(n)=nlgn

then $\log_b a = \log_4 3 = 0.8$ and f(n)=nlgn

f(n)=Ω(n) =Ω($n^{0.8+0.2}$) .

It matches case 3 when ε=0.2 and gives us:

T(n)=Θ(f(n))=Θ(nlgn)

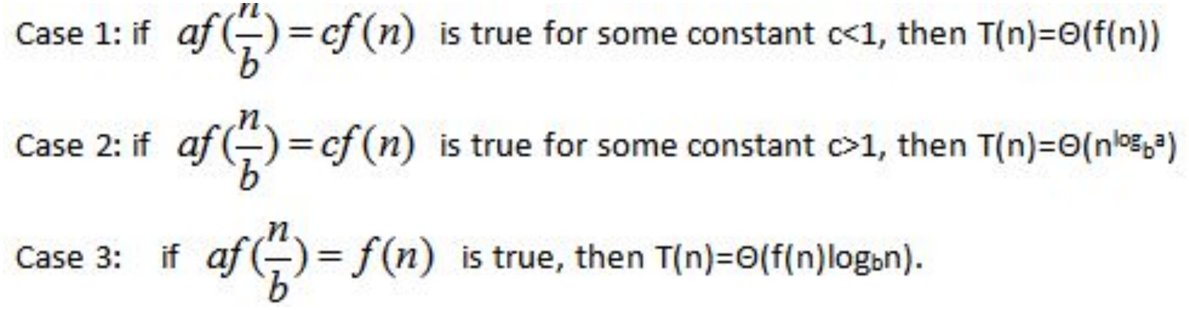
Ex: Solve T(n)'s asymptotic bound when

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

a=2, b=2, f(n)=n^2, then $n^{\log_b a} = n$, that violates the first two cases. (the upper bound of n^2 is impossible to be n or $n^{1-\varepsilon}$)

Let's try case 3.
Prove that f(n)=n^2=$\Omega(n^{1+\varepsilon})$.
We can find 1 as the value of ε satifying the above statement.
So it matches case 3 and gives us:
T(n)=Ө(f(n))=Ө(n^2)

Extended Form of Master Theorem



Case 1: if $af(\frac{n}{b}) = cf(n)$ is true for some constant $c<1$, then $T(n)=\Theta(f(n))$

Case 2: if $af(\frac{n}{b}) = cf(n)$ is true for some constant $c>1$, then $T(n)=\Theta(n^{\log_b a})$

Case 3: if $af(\frac{n}{b}) = f(n)$ is true, then $T(n)=\Theta(f(n)\log_b n)$.

Example---->

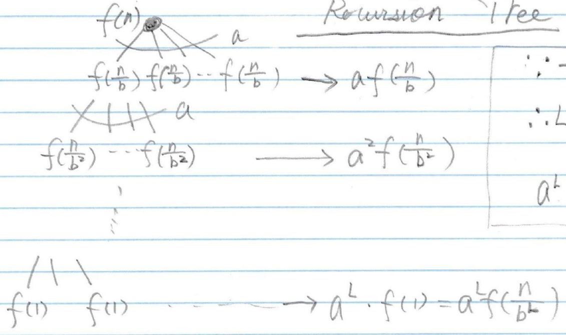
By Lan Yao

Proof

$T(n) = aT(\frac{n}{b}) + f(n)$

Prove Extented form of Master Theorem

Recursion Tree


$f(n)$
a
$f(\frac{n}{b})\ f(\frac{n}{b}) \cdots f(\frac{n}{b}) \longrightarrow af(\frac{n}{b})$
a
$f(\frac{n}{b^2}) \cdots f(\frac{n}{b^2}) \longrightarrow a^2 f(\frac{n}{b^2})$
$f(1)\quad f(1) \cdots \longrightarrow a^L \cdot f(1) = a^L f(\frac{n}{b^L})$

$\because \frac{n}{b^L} = 1 \Rightarrow$

$\therefore L = \log_b n$

$a^L = a^{\log_b n} = n^{\log_b a}$

▲ the number of levels: $L = \log_b n$

▲ the number of leaves: $a^L = a^{\log_b n} = n^{\log_b a}$

▲ Total cost $= f(n) + \underline{af(\frac{n}{b})} + \underline{a^2 f(\frac{n}{b^2})} + \cdots + a^L f(\frac{n}{b^L})$

suppose that $af(\frac{n}{b}) = cf(n)$, plus it in:

$af(\frac{n}{b}) = cf(n)$

$a^2 f(\frac{\frac{n}{b}}{b})$

$= a \cdot af(\frac{\frac{n}{b}}{b})$

$= a \cdot cf(\frac{n}{b})$

$= c \cdot af(\frac{n}{b})$

$= c \cdot cf(n)$

$= c^2 f(n)$

Then the total cost is:

$f(n) + cf(n) + c^2 f(n) + \cdots + c^L f(n)$

$= f(n)(1 + c + c^2 + c^L)$

$= f(n) \frac{c^{L+1} - 1}{c - 1}$

$$T(n) = f(n)\frac{c^{L+1}-1}{c-1} = f(n)(1+c+c^2+\cdots+c^L)$$

① when $c=1$, $T(n) = f(n)\cdot(L+1) = f(n)(\log_b n+1)$

when $n \to \infty$ $\Theta(\log_b n+1) = \Theta(\log_b n)$

$\therefore T(n) = \Theta(f(n)\cdot\log_b n)$

② when $c<1$, $T(n) = \Theta(f(n))$

③ when $c>1$, last level $= f(n)c^L$

$c^L = c^{\log_b n} = n^{\log_b c}$

from the recursion tree: $a^L f(1) = a^L\cdot K$

$\Rightarrow$

$f(n)\cdot c^L = f(n)\cdot n^{\log_b c} = a^{\log_b n}\cdot K = n^{\log_b a}\cdot K$

while $T(n) = \Theta(c^L f(n)) = \Theta(n^{\log_b a}\cdot K)$

$= \Theta(n^{\log_b a})$