ILLINOIS INSTITUTE OF TECHNOLOGY

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

CS430

Introduction to Algorithms

Lec 5

Lan Yao

Outlines

- Recursion Tree
- Master Theorem and Extended Form
- Selection Sort and Bubble Sort

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:

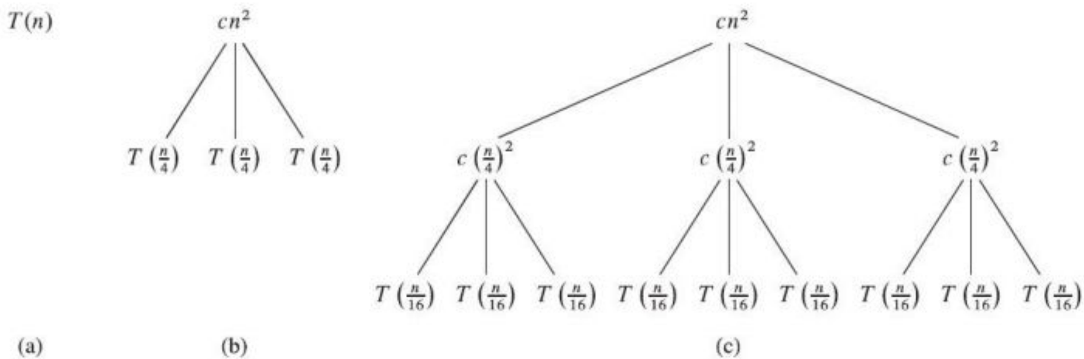
$$T(n) <= c\left\lfloor \frac{n}{2} \right\rfloor - d + c\left\lceil \frac{n}{2} \right\rceil - d + 1 = cn - 2d + 1 = cn - (2d - 1)$$

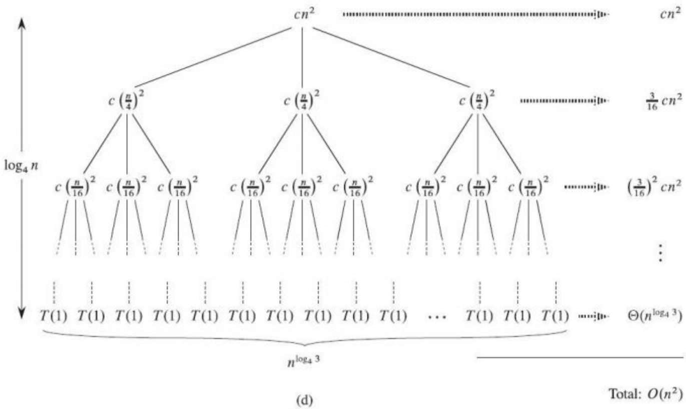
when 2d-1>d, T(n)<=cn-d.

(2d-1>=d=d>=1, which is easy to satisfy) QED.

Recursion Tree

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





(d)

- Number of levels L: L=$\log_4$n, for more accuracy, L=$\log_4$n+1
- Cost of the i^{th} 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 : $cn^{\log_4 3} = \Theta(n^{\log_4 3})$

Sum up all costs from all levels:

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

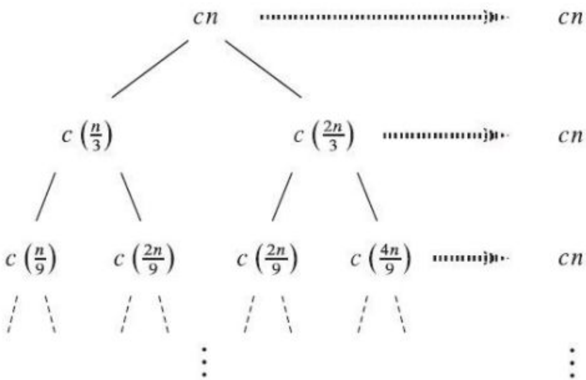
$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

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

Last example for Recursion Tree

EX2: Solve the upper bound of

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



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

then

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

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

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

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

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

$$\boxed{T(n)=8T(n/2)+n^2}$$

By Lan Yao

Ex: Solve the upper bound of

$T(n)=8T(n/2)+n^2$

Derive Master Theorem, then we have

$a=8$, $b=2$, $f(n)=n^2$

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

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

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

Case1: $T(n)=aT(n/b)+f(n)$

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

By Lan Yao

Case1: $T(n)=aT(n/b)+f(n)$

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

By Lan Yao

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)

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)

Case1:

If $f(n) = O(n^{\log_b a-\varepsilon})$

Case2:

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

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. (Because the upper bound of n^2 is impossible to be n or $n^{1-\varepsilon}$)

Case 3:

If $f(n) = \Omega(n^{\log_b a+\varepsilon})$

Let's try case 3.

Prove that f(n)=n^2=$\Omega(n^{1+\varepsilon})$.

We can find 1 as the value of ε satisfying the above statement.

So it matches case 3 and gives us:

T(n)=Θ(f(n))=$\Theta(n^2)$

Extended Form of Master Theorem

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

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

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

$\vdots$

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

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

$\therefore L = \log_b n$

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

- the number of levels: $L = \log_b n$
- the number of leaves: $a^L = a^{\log_b n} = n^{\log_b a}$
- Total cost $= f(n) + af(\frac{n}{b}) + a^2 f(\frac{n}{b^2}) + \cdots + a^L f(\frac{n}{b^L})$

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

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

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

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

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

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

$= c \cdot cf(n)$

$= c^2 f(n)$

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

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

Extended form of Master Theorem

Methodology

Step1: list values of a, b and f(n)

Step2: plug a and b in to evaluate af(n/b)

Step3: set $af(\frac{n}{b}) = cf(n)$ and solve the value of c.

Step4: match the value of c to the above 3 cases.

case1: when c<1, T(n)=Θ(f(n))
case 2: when c>1, T(n)=Θ(n^{logba})
case 3: when c=1, T(n)=Θ(f(n)$\log_b$n)

By Lan Yao

Ex1. Solve the asymptotic bound of T(n), when $T(n) = 3T(\frac{n}{2}) + n$

$a=3$, $b=2$, $f(n)=n$

$af(\frac{n}{b}) = 3f(\frac{n}{2}) = 3 \cdot \frac{n}{2} = \frac{3}{2}n$

Suppose: $\frac{3}{2}n = cf(n) = c \cdot n = cn$

then $c = \frac{3}{2} > 1 \rightarrow$ match

Case 2 gives you: $T(n) = \Theta(n^{\log_b a})$

$= \Theta(n^{\log_2 3})$

case1: when c<1, T(n)=Θ(f(n))
case 2: when c>1, $T(n)=\Theta(n^{\log_b a})$
case 3: when c=1, $T(n)=\Theta(f(n)\log_b n)$

By Lan Yao

Ex2. Solve the asymptotic bound of T(n), when $T(n) = T(\frac{3n}{4}) + n$

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

$a = 1,\ b = \frac{4}{3},\ f(n) = n$

$af(\frac{n}{b}) = 1 \cdot f(\frac{3}{4}n) = \frac{3}{4}n$

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

that is: $\frac{3}{4}n = c \cdot n \Rightarrow c = \frac{3}{4}$

$\because c = \frac{3}{4} < 1 \quad \therefore$ case 1

Case 1 gives us: $T(n) = \Theta(f(n))$

$= \Theta(n)$

case1: when c<1, $T(n)=\Theta(f(n))$
case 2: when c>1, $T(n)=\Theta(n^{\log_b a})$
case 3: when c=1, $T(n)=\Theta(f(n)\log_b n)$

Ex3. Solve the asymptotic bound of T(n), when $T(n) = 2T(\frac{n}{2}) + n$

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

$a = 2, \quad b = 2, \quad f(n) = n$

$af(\frac{n}{b}) = 2f(\frac{n}{2}) = 2 \cdot \frac{n}{2} = n$

suppose that $af(\frac{n}{b}) = cn \Rightarrow n = cn$

$\Rightarrow c = 1 \Rightarrow$ case 3

Case 3 gives us: $T(n) = \Theta(f(n) \cdot \log_b n)$

$= \Theta(n \lg n)$

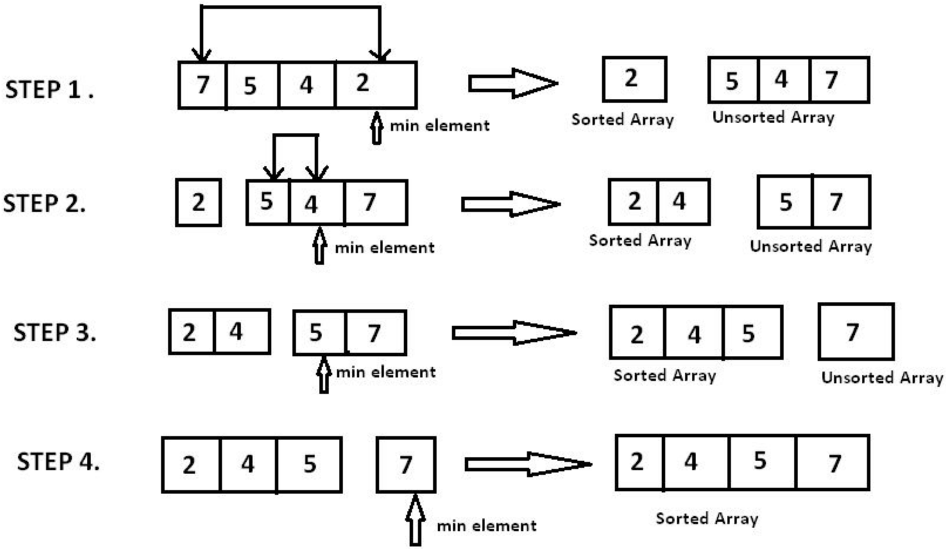
case1: when c<1, T(n)=Θ(f(n))
case 2: when c>1, T(n)=Θ(n^{logba})
case 3: when c=1, T(n)=Θ(f(n)$\log_b$n)

Algorithmic Analysis of other Sorts

- Selection Sort

- Bubble Sort

Selection Sort



Complexity of Selection Sort

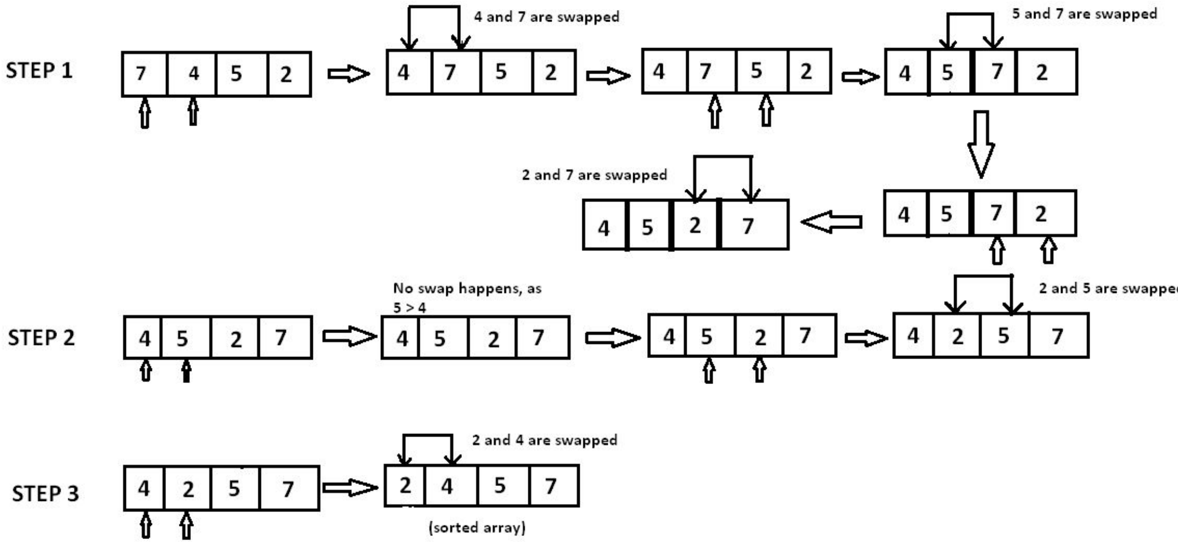
$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = O(n^2)$$

- 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
2 and 7 are swapped
4
5
2
7
4
5
7
2
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:

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

$$= O(n^2)$$