Proof of Master method

CS-6th

Instructor: Dr. Ayesha Enayet

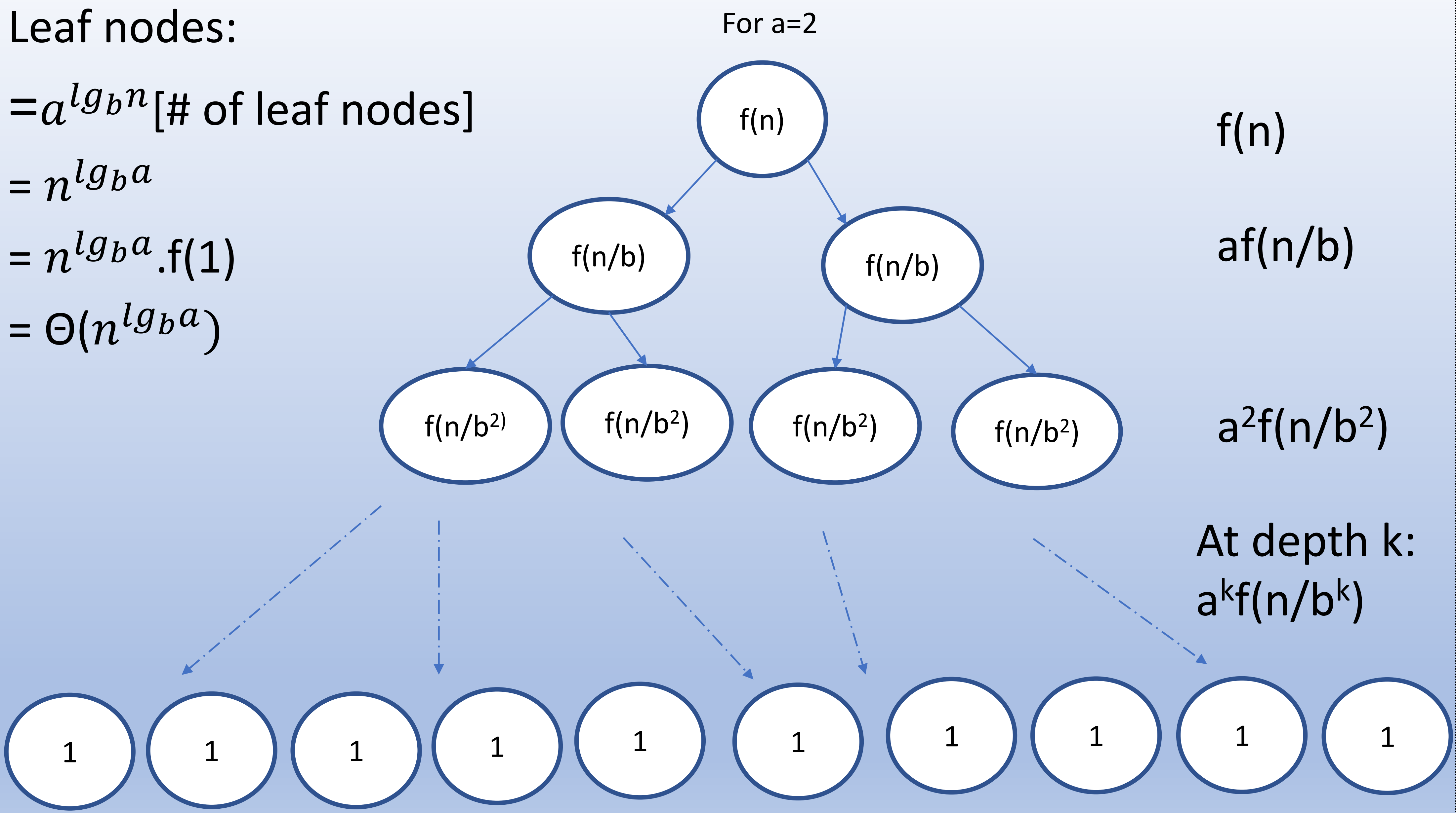
Leaf nodes:

$= a^{lg_b n}$[# of leaf nodes]

$= n^{lg_b a}$

$= n^{lg_b a}$.f(1)

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



f(n)

af(n/b)

$a^2f(n/b^2)$

At depth k:

$a^kf(n/b^k)$

Total Cost

- Total Cost=Cost at all internal nodes + leaf nodes
- T(n)= $\sum_{k=0}^{lg_b n - 1} a^k f(\frac{n}{b^k})$ +Θ($n^{lg_b a}$)→ eq1

Master Theorem

1. If there exists a constant $\epsilon > 0$ such that $f(n) = O(n^{\log_b a - \epsilon})$, then $T(n) = \Theta(n^{\log_b a})$.
2. If there exists a constant $k \geq 0$ such that $f(n) = \Theta(n^{\log_b a} \lg^k n)$, then $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$.
3. If there exists a constant $\epsilon > 0$ such that $f(n) = \Omega(n^{\log_b a + \epsilon})$, and if $f(n)$ additionally satisfies the ***regularity condition*** $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ∎

Case 1: f(n)< $n^{lg_b a}$

- T(n)= $\sum_{k=0}^{lg_b n-1} a^k f(\frac{n}{b^k})$ +Θ($n^{lg_b a}$)→ eq1
- By Case 1:
  - If f(n)= O($n^{lg_b a-\epsilon}$) $\text{implies} f(\frac{n}{b^k})$= O($\left(\frac{n}{b^k}\right)^{lg_b a-\epsilon}$)) substitute in eq1
- T(n)<= $\sum_{k=0}^{lg_b n-1} a^k \left(\frac{n}{b^k}\right)^{lg_b a-\epsilon}$ + Θ($n^{lg_b a}$)
- T(n)<= $n^{lg_b a-\epsilon} \sum_{k=0}^{lg_b n-1} a^k \left(\frac{1}{b^k}\right)^{lg_b a-\epsilon}$ + Θ($n^{lg_b a}$)
- T(n)<= $n^{lg_b a-\epsilon} \sum_{k=0}^{lg_b n-1} (a \left(\frac{b^{\epsilon}}{b^{lg_b a}}\right)$ $)^k$ + Θ($n^{lg_b a}$) $[by\ b^{lg_b a} =$ a]

- T(n)<= $n^{lg_b a-\epsilon} \sum_{k=0}^{lg_b n-1} (a\left(\frac{b^{\epsilon}}{b^{lg_b a}}\right)$)^k +Θ($n^{lg_b a}$) [$by\ b^{lg_b a}$ = a]
- T(n)<= $n^{lg_b a-\epsilon} \sum_{k=0}^{lg_b n-1} (a\left(\frac{b^{\epsilon}}{a}\right)$)^k +Θ($n^{lg_b a}$)
- T(n)<= $n^{lg_b a-\epsilon} \sum_{k=0}^{lg_b n-1} (b)^{\epsilon k}$ +Θ($n^{lg_b a}$)
- T(n)<= $n^{lg_b a-\epsilon} \sum_{k=0}^{lg_b n-1} (b^{\epsilon})^k$ +Θ($n^{lg_b a}$)

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

- T(n)<= $n^{lg_b a-\epsilon} \frac{b^{\epsilon lg_b n-1}-1}{b^{\epsilon}-1}$ + Θ($n^{lg_b a}$)[by geometric series]

- T(n)<= $n^{lg_b a-\epsilon}\frac{b^{\epsilon lg_b n}-1}{b^{\epsilon}-1}$ + $\Theta(n^{lg_b a})$
- T(n)<= $n^{lg_b a-\epsilon}\frac{n^{\epsilon}-1}{b^{\epsilon}-1}$ + $\Theta(n^{lg_b a})$[by disregarding contant terms b and ϵ]
- T(n)= $\Theta(n^{lg_b a})$

Master Theorem

1. If there exists a constant $\epsilon > 0$ such that $f(n) = O(n^{\log_b a - \epsilon})$, then $T(n) = \Theta(n^{\log_b a})$.

2. If there exists a constant $k \geq 0$ such that $f(n) = \Theta(n^{\log_b a} \lg^k n)$, then $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$.

3. If there exists a constant $\epsilon > 0$ such that $f(n) = \Omega(n^{\log_b a + \epsilon})$, and if $f(n)$ additionally satisfies the ***regularity condition*** $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ∎

Case 2: f(n)<= $n^{lg_b a}$

- T(n)= $\sum_{k=0}^{lg_b n-1} a^k \left(\frac{n}{b^k}\right)^{lg_b a}$ + Θ($n^{lg_b a}$) [substitute f$\left(\frac{n}{b^k}\right) =$ Θ(($\frac{n}{b^k}$)$^{lg_b a}$)]
- = $n^{lg_b a} \sum_{k=0}^{lg_b n-1} a^k \left(\frac{1}{b^k}\right)^{lg_b a}$ + Θ($n^{lg_b a}$)
- =$n^{lg_b a} \sum_{k=0}^{lg_b n-1} a^k \left(b^{-k}\right)^{lg_b a}$ + Θ($n^{lg_b a}$) [$by\ b^{lg_b a} =$ a]
- = $n^{lg_b a} \sum_{k=0}^{lg_b n-1} a^k \left(a^{-k}\right)$ + Θ($n^{lg_b a}$)
- =$n^{lg_b a}(lg_b n + 1)$ + Θ($n^{lg_b a}$) [take n-1≈n]
- = Θ($n^{lg_b a}\ lg_b n$)

Master Theorem

1. If there exists a constant $\epsilon > 0$ such that $f(n) = O(n^{\log_b a - \epsilon})$, then $T(n) = \Theta(n^{\log_b a})$.

2. If there exists a constant $k \geq 0$ such that $f(n) = \Theta(n^{\log_b a} \lg^k n)$, then $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$.

3. If there exists a constant $\epsilon > 0$ such that $f(n) = \Omega(n^{\log_b a + \epsilon})$, and if $f(n)$ additionally satisfies the ***regularity condition*** $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ∎

Case 3: f(n)> $n^{lg_b a}$

- f(n)=Ω ($n^{lg_b\, a+\epsilon}$)for any ϵ>0 with c= $b^{-\epsilon}$<1 and af($\frac{n}{b}$)<=cf(n)
- af($\frac{n}{b}$)=a$(\frac{n}{b})^{lg_b\, a+\epsilon}$=a$n^{lg_b\, a+\epsilon}$ $b^{-lg_b\, a}b^{-\epsilon} = a\,\frac{n^{lg_b\, a+\epsilon}}{b^{lg_b\, a}}\, b^{-\epsilon}$
- = $n^{lg_b\, a+\epsilon}$ $b^{-\epsilon}$ =f(n) $b^{-\epsilon}$
- a^kf($\frac{n}{b^k}$)<=c^kf(n) [af($\frac{n}{b}$)<=cf(n)]
- T(n)= $\sum_{k=0}^{lg_b\, n-1} a^k f(\frac{n}{b^k})$ +Θ($n^{lg_b a}$)→ eq1
- T(n)<= $\sum_{k=0}^{lg_b\, n-1} c^k$f(n) +Θ($n^{lg_b a}$)

- T(n)<= f(n)$\sum_{k=0}^{\infty} c^k$ +Θ($n^{lg_b a}$)
- = Θ($n^{lg_b a}$)+ f(n)$\frac{1}{1-c}$ [Sum of infinite geometric series a+ar+ar^2+ar^3... = a / (1 − r)
- = Θ(f(n))

Important concept

- What this sum looks like depends on how the asymptotic growth of *f(n)* compares to the asymptotic growth of the number of leaves. There are three cases:
- Case 1: $f(n)$ is $O(n^{\log_b a - \varepsilon})$. Since the leaves grow faster than *f*, asymptotically all of the work is done at the leaves, so $T(n)$ is $\Theta(n^{\log_b a})$.
- Case 2: $f(n)$ is $\Theta(n^{\log_b a})$. The leaves grow at the same rate as *f*, so the same order of work is done at every level of the tree. The tree has O(log n) levels, times the work done on one level, yielding $T(n)$ is $\Theta(n^{\log_b a} \log n)$.
- Case 3: $f(n)$ is $\Omega(n^{\log_b a + \varepsilon})$. In this case *f* grows faster than the number of leaves, which means that asymptotically the total amount of work is dominated by the work done at the root node. For the upper bound, we also need an extra smoothness condition on *f* in this case, namely that $af(n/b) \leq cf(n)$ for some constant $c < 1$ and large *n*. In this case $T(n)$ is $\Theta(f(n))$.

[Reference: https://www.cs.cornell.edu/courses/cs3110/2011sp/Lectures/lec19-master/master.htm]