4.4 Proof of the master theorem

- Not the full proof: only the intuition
- Analyzes the master recurrence

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

for exact powers of $b > 1$, i.e. $n = b^1, b^2, b^3, \ldots$

- This is enough to see why the master theorem is true
- Full proof: $\begin{cases} n \in \mathbb{N} \text{ instead of } n = b^1, b^2, b^3, \ldots \\ \text{floors} \\ \text{ceilings} \end{cases}$

Notes:

Master Theorem:

$f(n) = O(n^{\log_b a - \varepsilon}) \Rightarrow T(n) = \Theta(n^{\log_b a}$

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

$f(n) = \Omega(n^{\log_b a + \varepsilon}) \Rightarrow T(n) = \Theta(f(n))$

4.4.1 The proof for exact powers

- Analysis of the recurrence

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

under the assumption that n is an exact power of $b > 1$ where b need not to be an integer

- Analysis is broken into three lemmas:

  First lemma: reduces the problem of solving the master recurrence to the problem of evaluating an expression that contains a summation.

  Second lemma: determines bounds on this summation

  Third lemma: puts the first two together to prove a version of the master theorem for the case in which n is an exact power of b.

Lemma 4.2

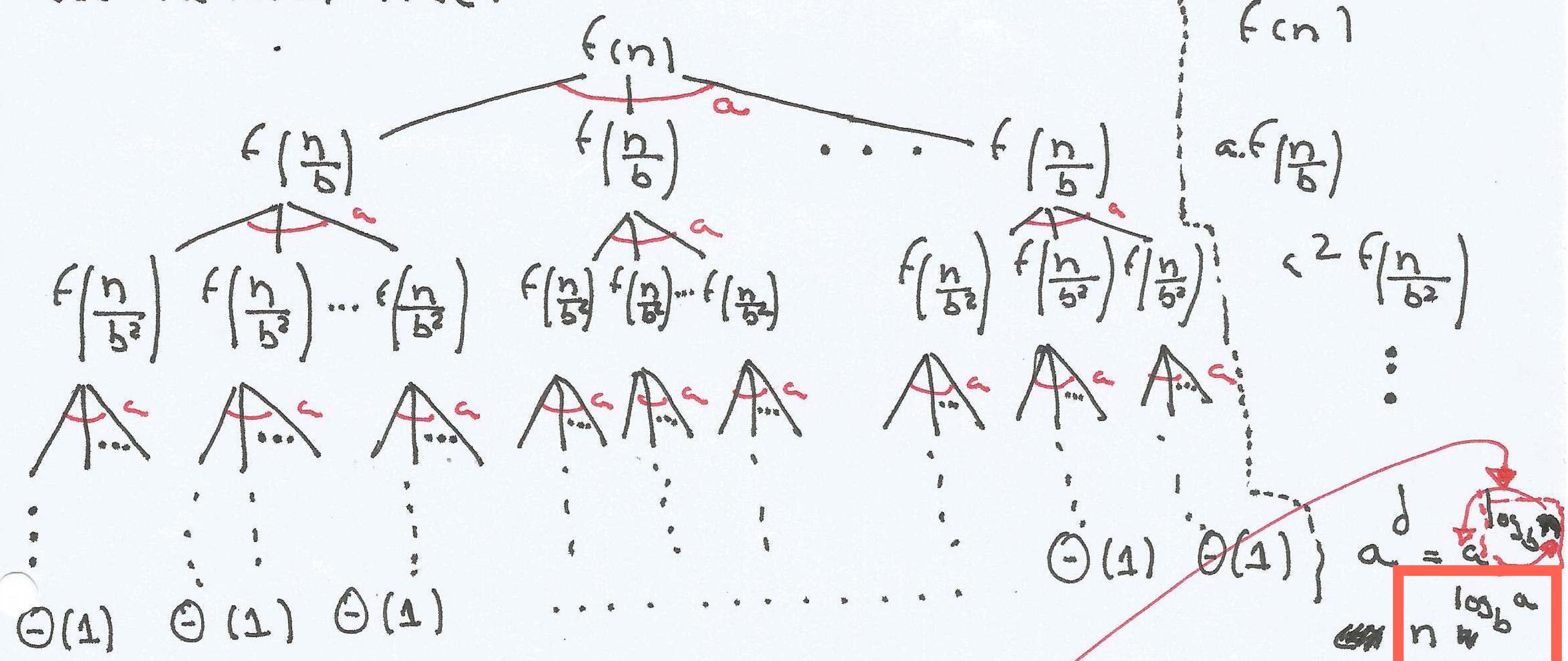
- Let $a \geq 1$ and $b > 1$ be constants, and let $f(n)$ be a nonnegative function defined on exact powers of b.
- Define $T(n)$ on exact powers of b by the recurrence:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ aT\left(\frac{n}{b}\right) + f(n) & \text{if } n = b^i,\ \forall i \in \mathbb{N}^+ \end{cases}$$

- Then $$\boxed{T(n) = \Theta\left(n^{\log_b a}\right) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)}$$ Expression: 4.6

Proof:

- Use recursion tree:



Here should be $\Theta{n^{log_{b}{a}}}$

Question: What is the tree depth?

$$\left.\begin{array}{ll} 0^{th} \text{ level}: & n \\ 1^{st} \text{ level}: & \frac{n}{b} \\ 2^{nd} \text{ level}: & \frac{n}{b^2} \\ \vdots & \\ d^{th} \text{ level}: & \frac{n}{b^d} \end{array}\right\} d = \log_b n$$

Question:

$\boxed{\text{What is the cost per level?}}$

Level 0: $f(n) = a^0 \cdot f\left(\frac{n}{b^0}\right)$

level 1: $a^1 f\left(\frac{n}{b^1}\right)$

level 2: $a^2 f\left(\frac{n}{b^2}\right)$

$\vdots$

level j: $\boxed{a^j f\left(\frac{n}{b^j}\right)}$

Question:

$\boxed{\text{What is the total cost of the tree?}}$

$$T(n) = \sum_{j=0}^{d-1} a^j f\left(\frac{n}{b^j}\right) + n^{\log_b a}$$

$d = \log_b n$

$$= \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) + n^{\log_b a}$$

Question:

What can be said about this summation?

■ (end of proof of lemma 4.2)

Lemma 4.3

- Let $a \geq 1$ and $b > 1$ be constants, and let $f(n)$ be a nonnegative function defined on exact powers of b
- A function $g(n)$ defined over exact powers of b by

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) \quad \text{(equation 4.7)}$$

can then be bounded asymptotically on exact powers of b as follows (continues in the next page)

case 1. If $f(n) = O\left(n^{\log_b a - \varepsilon}\right)$ for some constant $\varepsilon > 0$

then $g(n) = O\left(n^{\log_b a}\right)$

case 2. If $f(n) = \Theta\left(n^{\log_b a}\right)$ then $g(n) = \Theta\left(n^{\log_b a} \log n\right)$

case 3. If $a f\left(\frac{n}{b}\right) \leq c f(n)$ for some constant $c < 1$ and

for all $n \geq b$ then $g(n) = \Theta(f(n))$

Proof

- Proof of case 1

- $f(n) = O\left(n^{\log_b a - \varepsilon}\right) \Rightarrow f\left(\frac{n}{b^j}\right) = O\left(\left(\frac{n}{b^j}\right)^{\log_b a - \varepsilon}\right)$

- Substituting into Equation 4.7 yields:

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) = \sum_{j=0}^{\log_b n - 1} a^j O\left(\left(\frac{n}{b^j}\right)^{\log_b a - \varepsilon}\right)$$

$$= O\left(\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a - \varepsilon}\right)$$

- $$\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a - \varepsilon} = \sum_{j=0}^{\log_b n - 1} a^j \frac{n^{\log_b a - \varepsilon}}{b^{j \cdot \log_b a - \varepsilon}} =$$

Factor out terms

$$= n^{\log_b a - \varepsilon} \cdot \sum_{j=0}^{\log_b n - 1} \left(\frac{a}{b^{\log_b a - \varepsilon}}\right)^j = n^{\log_b a - \varepsilon} \cdot \sum_{j=0}^{\log_b n - 1} \left(\frac{\cancel{a}}{\underbrace{b^{\log_b a}}_{\cancel{a}} \cdot b^{-\varepsilon}}\right)^j$$

$$= n^{\log_b a - \varepsilon} \cdot \sum_{j=0}^{\log_b n - 1} \left(b^{\varepsilon}\right)^j = n^{\log_b a - \varepsilon} \left(\frac{b^{\varepsilon \log_b n} - 1}{b^{\varepsilon} - 1}\right) =$$

$$n^{\log_b a - \varepsilon} \left(\frac{\left(\overbrace{b^{\log_b n}}^{n}\right)^{\varepsilon} - 1}{b^{\varepsilon} - 1}\right)$$

$$= n^{\log_b a - \varepsilon} \left(\frac{n^{\varepsilon} - 1}{b^{\varepsilon} - 1}\right)$$

$\boxed{\text{Notes:}}$

Geometric series:

$$\sum_{k=0}^{n} q^k = \frac{q^{n+1} - 1}{q - 1}$$

- Since b and ε are constants we can rewrite the last expression as:

$$n^{\log_b a - \varepsilon}\left(\frac{n^{\varepsilon} - 1}{\underbrace{b^{\varepsilon} - 1}_{\text{"constant"}}}\right) = n^{\log_b a - \varepsilon}\underbrace{O(n^{\varepsilon})}_{\text{which implies}} \overset{\text{Overall:}}{=} O\left(n^{\log_b a}\right)$$

- Substituting this expression into:

$$g(n) = O\left(\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a - \varepsilon}\right)$$

$$= O\left(n^{\log_b a}\right)$$

and case 1 is proved.

Proof of case 2:

· Under the assumption that $f(n) = \Theta(n^{\log_b a})$ we have that:

$$f(n) = \Theta(n^{\log_b a}) \Rightarrow f\left(\frac{n}{b^j}\right) = \Theta\left(\left(\frac{n}{b^j}\right)^{\log_b a}\right)$$

Substituting into Equation 4.7 yields:

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) = \sum_{j=0}^{\log_b n - 1} a^j \Theta\left(\left(\frac{n}{b^j}\right)^{\log_b a}\right)$$

$$= \Theta\left(\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}\right)$$

- We bound the summation within Θ as in case 1, but this time we do not obtain a geometric series

- Instead we discover that every term of the summation is the same:

$$\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a} = \sum_{j=0}^{\log_b n - 1} a^j \frac{n^{\log_b a}}{b^{j \log_b a}} = n^{\log_b a} \sum_{j=0}^{\log_b n - 1} \Bigg(\frac{\cancel{a}}{\underbrace{b^{\log_b a}}_{\cancel{a}}}\Bigg)^j$$

$$= n^{\log_b a} \sum_{j=0}^{\log_b n - 1} 1^j = \boxed{n^{\log_b a} \log_b n}$$

- Substituting this expression for the summation yields:

$$g(n) = \Theta\left(\sum_{j=0}^{\log_b n - 1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}\right)$$

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

and case 2 is proved

Proof of case 3:

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) \quad \text{(Expression 4.7)}$$

Since:

- $f(n)$ appears in the definition of $g(n)$
- All terms of $g(n)$ are nonnegative

$\Rightarrow g(n) = \Omega(f(n))$ for exact powers of $\underline{b}$

Under our assumption that:

$$a f\left(\frac{n}{b}\right) \leq c f(n)$$

for some constant $\underline{c > 1}$ and all $n \geq b$

$$\Leftrightarrow f\left(\frac{n}{b}\right) \leq \frac{c}{a} f(n)$$

Iterating $\underline{j}$ times: $f\left(\frac{n}{b^j}\right) \leq \left(\frac{c}{a}\right)^j f(n) \Leftrightarrow a^j f\left(\frac{n}{b^j}\right) \leq c^j f(n)$

Notes:

From a different perspective:

$a f\left(\frac{n}{b}\right) \leq c f(n)$ Iterate j times, i.e. draw a recursion tree of depth-level j

$$a^j f\left(\frac{n}{b^j}\right) \leq c^j f\left(\frac{n}{1^j}\right) \Leftrightarrow a^j f\left(\frac{n}{b^j}\right) \leq c^j f(n) \Leftrightarrow$$

$$\Leftrightarrow f\left(\frac{n}{b^j}\right) \leq \left(\frac{c}{a}\right)^j f(n)$$

Substituting into Equation 4.7 yields:

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) \le \sum_{j=0}^{\log_b n - 1} c^j f(n)$$

$$= f(n) \sum_{j=0}^{\log_b n - 1} c^j \le f(n) \underbrace{\sum_{j=0}^{\infty} c^j}$$

Notes:

Geometric series:

$$\sum_{k=0}^{\infty} a^k = \frac{1}{1-a}, \text{ if } |a| < 1$$

$$= f(n) \left(\frac{1}{1-c}\right) = \underline{\Theta(f(n))}$$

This is constant → Therefore

and case 3 is proved

Since:

$$\left.\begin{array}{l} g(n) = \Omega(f(n)) \\ g(n) = O(f(n)) \end{array}\right\} g(n) = \Theta(f(n))$$

We can now prove a version of the master theorem for the case in which $\underline{n}$ is an exact power of $\underline{b}$.

Lemma 4.4

- Let $a \geq 1$ and $b > 1$ be constants, and let $f(n)$ be a nonnegative function defined on exact powers of $\underline{b}$. Define $T(n)$ on exact powers of $\underline{b}$ by the recurrence:

$$T(n) = \begin{cases} \Theta(1) & , \text{ if } n = 1 \\ a\,T\left(\frac{n}{b}\right) + f(n), & \text{if } n = b^j \end{cases}$$

Where $\underline{i}$ is a positive integer

- Where i is a positive integer.
- Then $T(n)$ can be bounded asymptotically for exact powers of b as follows:

1. If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, $T(n) = \Theta(n^{\log_b a})$
2. If $f(n) = \Theta(n^{\log_b a})$ then $T(n) = \Theta(n^{\log_b a} \lg n)$
3. If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ ~~then~~ for some constant $\varepsilon > 0$ and if $a f\left(\frac{n}{b}\right) \leq c f(n)$ for some constant $c < 1$ and all sufficiently large n then $T(n) = \Theta(f(n))$

Proof:

- We use the bounds in Lemma 4.3 to evaluate the summation 4.6 from Lemma 4.2
- Case 1: $T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right)$ [Summation 4.6 from Lemma 4.2]

$$= \Theta(n^{\log_b a}) + O(n^{\log_b a}) \quad \text{[bound from Lemma 4.3]}$$

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

Question:

- Why $\Theta(n^{\log_b a})$?
  - $\Theta(n^{\log_b a})$ also include $O(n^{\log_b a})$ therefore the term $\boxed{O(n^{\log_b a})}$ is not adding anything

- <u>case 2</u>:

$$T(n) = \Theta\left(n^{\log_b a}\right) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) \quad \left[\text{Summation 4.6 from Lemma 4.2}\right]$$

$$= \Theta\left(n^{\log_b a}\right) + \Theta\left(n^{\log_b a} \log n\right) \quad \left[\text{bound from Lemma 4.3}\right]$$

$$= \Theta\left(n^{\log_b a} \log n\right) \quad \left(\text{greatest expression of the two when } n \to \infty\right)$$

- <u>case 3</u>:

$$T(n) = \Theta\left(n^{\log_b a}\right) + \sum_{j=0}^{\log_b n - 1} a^j f\left(\frac{n}{b^j}\right) \quad \left[\text{Summation 4.6 from Lemma 4.2}\right]$$

$$= \Theta\left(n^{\log_b a}\right) + \Theta\left(f(n)\right) \quad \left[\text{bound from Lemma 4.3}\right]$$

$$= \Theta\left(f(n)\right) \text{ because } f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$$

When n tends to infinity:

- because f(n) is $\Omega(n^{log_{b}{a + \epsilon}})$, i.e. $f(n)$ is at least n^{log_{b}{a + \epsilon}}

- we will have two expressions:

  - $\Theta(n^{\log_{b}{a}})$ and…

  - $\Omega(n^{log_{b}{a + \epsilon}})$

- Notice that $n^{log_{b}{a + \epsilon}}$ dominates over $n^{\log_{b}{a}}$ when n tends to infinity.

- This means that overall complexity is $\Theta(f(n))$

4.4.2 Floors and Ceilings

- To complete the proof of the master theorem we must extend our analysis to the situation where floors and ceilings are used
- I.e. recurrence defined for all integers, not just exact powers of b.

- Obtaining a lower bound on:

$$T(n) = aT(\underbrace{\lceil n/b \rceil}_{\geq n/b}) + f(n)$$

Is not interesting since $\lceil n/b \rceil \geq n/b$

- Also, obtaining an upper bound on

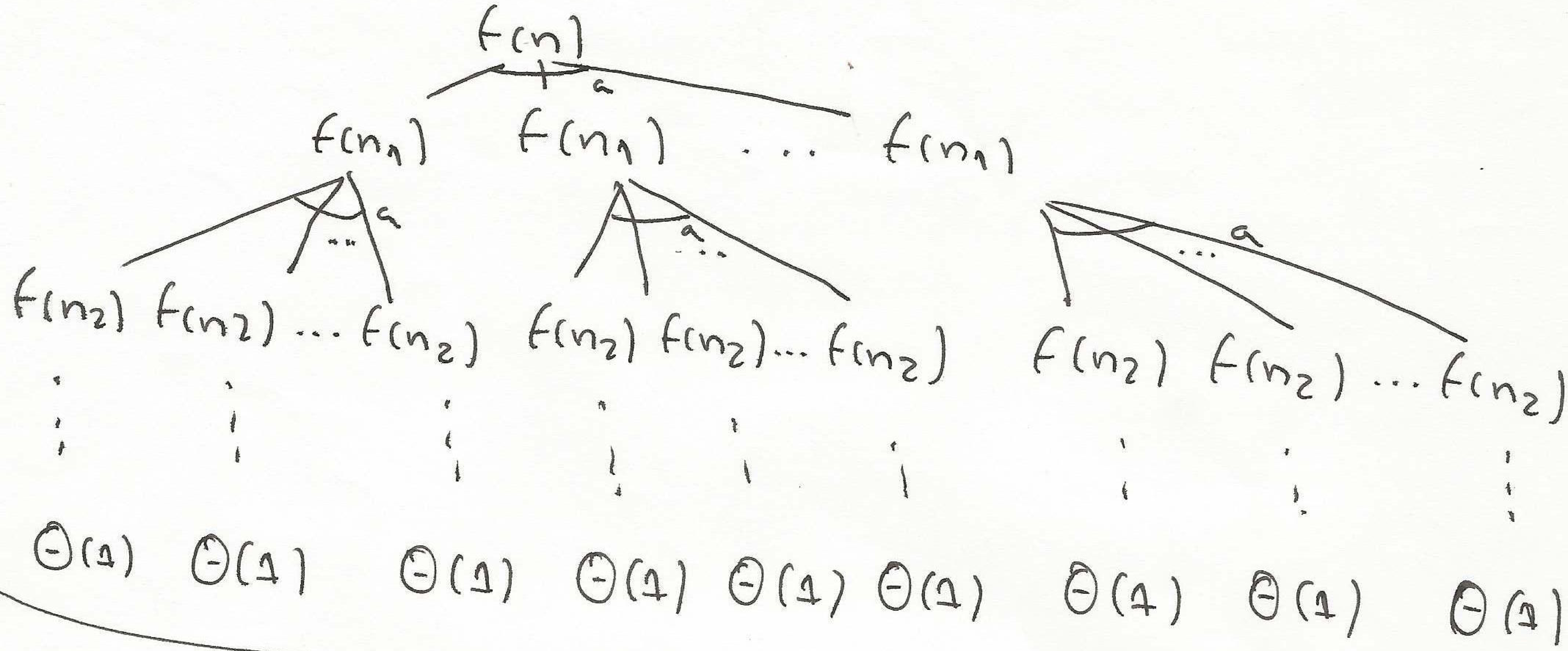
$$T(n) = aT(\lfloor n/b \rfloor) + f(n)$$

Is not interesting since $\lfloor n/b \rfloor \leq n/b$

- What is interesting?
  - Instead of calculating the lower-bound calculate the upper-bound
  - " " " " upper-bound " " lower-bound

→ ...

Recurrence tree:



Attention:

We are only focusing on upper bounding the recurrence

$$T(n) = a\,T(\lceil n/b \rceil) + f(n)$$

Cost per level:

level 0: $f(n)$

level 1: $a f(n_1)$

level 2: $a^2 f(n_2)$

$\vdots$

level k: $a^k f(n_k)$

- As we go down in the recursion tree, we obtain a sequence of recursive invocations on the arguments:

$$n$$
$$\lceil n/b \rceil$$
$$\lceil \lceil n/b \rceil / b \rceil$$
$$\lceil \lceil \lceil n/b \rceil / b \rceil / b \rceil$$
$$\vdots$$

- Let us denote the j^{th} element on the sequence by n_j where:

$$n_j = \begin{cases} n, & \text{if } j = 0 \\ \lceil n_{j-1}/b \rceil, & \text{if } j > 0 \end{cases}$$

(recursive definition)

- Our first goal: determine the depth k such that n_k is a constant

Why?

Remember: In a recurrence equation there will always be a base case that is applied when $n =$ some constant

- Using the inequality $\lceil u \rceil \le u + 1$ we obtain:

$$n_0 \le n$$

$$n_1 \le \left\lceil \frac{n_0}{b} \right\rceil \le \frac{n}{b} + 1$$

$$n_2 \le \left\lceil \frac{n_1}{b} \right\rceil \le \left\lceil \frac{\frac{n}{b} + 1}{b} \right\rceil \le \frac{n}{b^2} + \frac{1}{b} + 1$$

$$n_3 \le \left\lceil \frac{n_2}{b} \right\rceil \le \left\lceil \frac{\frac{n}{b^2} + \frac{1}{b} + 1}{b} \right\rceil \le \frac{n}{b^3} + \frac{1}{b^2} + \frac{1}{b} + 1$$

$$\vdots$$

In general:

$$n_j \leq \frac{n}{b^j} + \sum_{i=0}^{j-1} \frac{1}{b^i} < \frac{n}{b^j} + \sum_{i=0}^{\infty} \frac{1}{b^i}$$

$$= \frac{n}{b^j} + \frac{b}{b-1}$$

Notes:

$$\boxed{\sum_{i=0}^{\infty} b^{-i} = \frac{b}{b-1}}$$

Lets say that the tree height is $\lfloor \log_b n \rfloor$ then:

$$n_{\lfloor \log_b n \rfloor} < \frac{n}{b^{\lfloor \log_b n \rfloor}} + \frac{b}{b-1} \leq \frac{n}{b^{\log_b n - 1}} + \frac{b}{b-1}$$

$$= \frac{n}{b^{\log_b n} \cdot b^{-1}} + \frac{b}{b-1} = \frac{\cancel{n}}{n \cdot b^{-1}} + \frac{b}{b-1} =$$

$$= b + \frac{b}{b-1} = O(1)$$

- Thus at depth $\lfloor \log_b n \rfloor$ the problem size is at most constant
- I.e. tree height $= \lfloor \log_b n \rfloor$

- We are now able to calculate the total cost:

$$T(n) = \underbrace{\Theta(n^{\log_b a})}_{\text{number of leafs}} + \underbrace{\sum_{j=0}^{\lfloor \log_b n \rfloor - 1} a^j f(n_j)}_{\text{everything else}}$$

- We have already seen this equation when we were doing the proof for exact powers
- However, in this case we are not restricted to an exact power of b.
- Therefore we can follow the same procedure of Lemma 4.3

∎