COMP2230/COMP6230
Algorithms

Lecture 3

Professor Ljiljana Brankovic

Lecture Overview

- Recurrence Relations
- Analysis of Algorithms

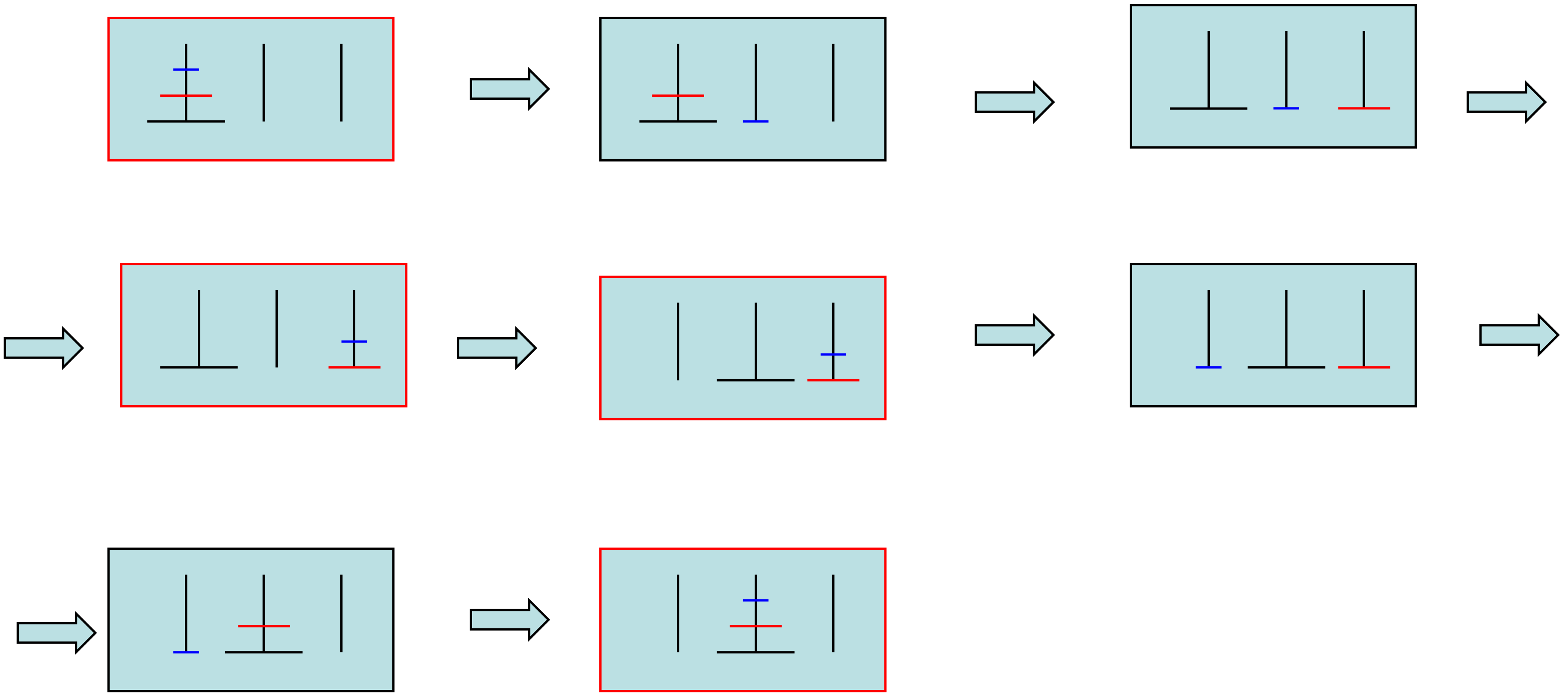
Example 1: Towers of Hanoi

After creating the world, God set on Earth 3 diamond rods and 64 golden rings, all of different size. All the rings were initially on the first rod, in order of size, the smallest at the top. God also created a monastery nearby where monks' task in life is to transfer all the rings onto the second rod; they are only allowed to move a single ring from one rod to another at the time, and a ring can never be placed on top of another smaller ring. According to the legend, when monks have finished their task, the world will come to an end.

If monks move one ring per second and never stop, it will take them more that 500,000 million years to finish the job (more than 35 times the estimated age of the universe!)

Example 1: Towers of Hanoi

The following is a way to move 3 rings from rod *1* to rod 2.



Example 1: Towers of Hanoi

The following is an algorithm that moves m rings from rod i to rod j.

```
Hanoi (m,i,j){
   \\Moves the m smallest rings from rod i to rod j
   if m > 0 then {Hanoi(m-1,i,6-i-j)
                  write i " →" j
                  Hanoi(m-1,6-i-j,j)}
}
```

(Note that $i + j + k = 6$, so the third rod k can be expressed as $k = 6 - i - j$)

Example 1: Towers of Hanoi

Recurrence relation:

$$t(m) = \begin{cases} 0 & \text{if } m = 0 \\ 2t(m-1)+1 & \text{otherwise} \end{cases}$$

or, equivalently,

$t_0 = 0$

$t_n = 2t_{n-1} + 1, n > 0$

Example 2

Show that for any real a and b, $b > 0$, $(n+a)^b = \Theta(n^b)$

$\boldsymbol{O}$: Show that there are constants C_1 and N_1 such that

$(n+a)^b \leq C_1 n^b$, for all $n \geq N_1$

$(n+a)^b \leq (2n)^b = 2^b n^b$ for all $n \geq a$

$\boldsymbol{\Omega}$: Show that there are constants C_2 and N_2 such that
$(n+a)^b \geq C_2 n^b$, for all $n \geq N_2$

$(n+a)^b \geq (\frac{n}{2})^b = (\frac{1}{2})^b n^b$ for all $n \geq 2|a|$

Example 3

When is $f(2n) = \Theta(f(n))$?

$f(n) = n^2$

O: $(2n)^2 \leq 4n^2$, for all $n \geq 0$

Ω: $(2n)^2 \leq n^2$ for all $n \geq 0$

$f(n) = 2^n$

O: $2^{2n} \leq C_1 2^n$, for all $n \geq N_1$???

$2^n 2^n \leq C_1 2^n$???

NO!!! There is no constant C_1 such that $2^{2n} \leq C_1$ for all $n \geq N_1$

Solving Recurrence Relations

Iteration (substitution): Example from first week's lecture

$C(n) = n + C(\lfloor \frac{n}{2} \rfloor), n > 1$

$C(1) = 0$

Solution for $n = 2^k$, for some k.

$$
\begin{aligned}
C(2^k) &= 2^k + C(2^{k-1}) \\
&= 2^k + 2^{k-1} + C(2^{k-2}) \\
&\dots \\
&= 2^k + 2^{k-1} + \dots + 2^1 + C(2^0) \\
&= 2^k + 2^{k-1} + \dots + 2^1 + C(1) \\
&= 2^k + 2^{k-1} + \dots + 2^1 + 0 \qquad \text{(as } C(1) = 0\text{)} \\
&= 2^{k+1} - 2 = 2n - 2 = \Theta(n).
\end{aligned}
$$

Solving Recurrence Relations

Solving the same recurrence for any n:

$$C(n) = n + C\left(\left\lfloor \frac{n}{2} \right\rfloor\right), \qquad n > 1$$

$$C(1) = 0$$

Solution for $2^{k-1} \leq n < 2^k$, for some k:

Since $C(n)$ is an increasing function (prove it!):

$C(2^{k-1}) \leq C(n) < C(2^k)$

$C(2^k) = 2^{k+1} - 2, C(2^{k-1}) = 2^k - 2$

$C(n) < C(2^k) = 2^{k+1} - 2 = 4 \times 2^{k-1} - 2 \leq 4n - 2 < 4n$, thus $C(n) = O(n)$

$C(n) \geq C(2^{k-1}) = 2^k - 2 > n - 2 \geq \frac{n}{2}$, for all $n \geq 4$, thus $C(n) = \Omega(n)$

Therefore, $C(n) = \Theta(n)$.

Main Recurrence Theorem (also known as Master Theorem)

Let a, b and k be integers satisfying $a \geq 1,\ b \geq 2$ and $k \geq 0$.

In the following, $\frac{n}{b}$ denotes either $\left\lfloor \frac{n}{b} \right\rfloor$ or $\left\lceil \frac{n}{b} \right\rceil$.

In the case of the floor function the initial condition $T(0) = u$ is given; and in the case of the ceiling function, the initial condition $T(1) = u$ is given.

Recurrence Theorem (also known as Master Theorem)

Upper Bound

If $T(n) \leq aT\left(\frac{n}{b}\right) + f(n)$ and $f(n) = O(n^k)$ then

$$T(n) = \begin{cases} O\left(n^k\right) & \text{if } a < b^k \\ O\left(n^k \log n\right) & \text{if } a = b^k \\ O\left(n^{\log_b a}\right) & \text{if } a > b^k \end{cases}$$

The same theorem holds for Ω and Θ notations as well.

Recurrence Theorem (also known as Master Theorem)

Lower Bound

If $T(n) \geq aT(\frac{n}{b}) + f(n)$ and $f(n) = \Omega(n^k)$ then

$$T(n) = \begin{cases} \Omega(n^k) & \text{if } a < b^k \\ \Omega(n^k \log n) & \text{if } a = b^k \\ \Omega(n^{\log_b a}) & \text{if } a > b^k \end{cases}$$

Recurrence Theorem (also known as Master Theorem)

Tight Bound

If $T(n) = aT(\frac{n}{b}) + f(n)$ and $f(n) = \Theta(n^k)$ then

$$T(n) = \begin{cases} \Theta\left(n^k\right) & \text{if } a < b^k \\ \Theta\left(n^k \log n\right) & \text{if } a = b^k \\ \Theta\left(n^{\log_b a}\right) & \text{if } a > b^k \end{cases}$$

Example 4

Solve $c_n = n + c_{\lfloor \frac{n}{2} \rfloor}, \ n > 1; c_1 = 0$

We have $a = 1, b = 2, f(n) = n$ and $k = 1.$

Since $a < b^{k}, c_n = \Theta(n^k)$ *and since* $k = 1, \ c_n = \Theta(n).$

If $T(n) = aT(\frac{n}{b}) + f(n)$ and $f(n) = \Theta(n^{k})$ then

$$T(n) = \begin{cases} \Theta\left(n^k\right) & \text{if } \ a < b^k \\ \Theta\left(n^k \log n\right) & \text{if } \ a = b^k \\ \Theta\left(n^{\log_b a}\right) & \text{if } \ a > b^k \end{cases}$$

Example 5

Solve $c_n = n + 2c_{\lfloor \frac{n}{2} \rfloor}, \ n > 1; c_1 = 0$

We have $a = 2, b = 2, f(n) = n$ and $k = 1$.

Since $a = b^k$, we have $c_n = \Theta(n^k \log n)$

and since $k = 1$, *we have* $c_n = \Theta(n \log n)$.

If $T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$ and $f(n) = \Theta(n^k)$ then

$$T(n) = \begin{cases} \Theta\left(n^k\right) & \text{if} \quad a < b^k \\ \Theta\left(n^k \log n\right) & \text{if} \quad a = b^k \\ \Theta\left(n^{\log_b a}\right) & \text{if} \quad a > b^k \end{cases}$$

Example 6

Solve $c_n = n + 7\, c_{\left\lfloor \frac{n}{4} \right\rfloor}$

We have $a = 7, b = 4, f(n) = n$ and $k = 1$.

Since $a > b^k$, $c_n = \Theta(n^{\log_4 7})$

If $\mathrm{T(n)} = \mathrm{aT}(\frac{n}{b}) + \mathrm{f(n)}$ and $\mathrm{f(n)} = \Theta(\mathrm{n}^{\mathrm{k}})$ then

$$T(n) = \begin{cases} \Theta\left(n^k\right) & \text{if } a < b^k \\ \Theta\left(n^k \log n\right) & \text{if } a = b^k \\ \Theta\left(n^{\log_b a}\right) & \text{if } a > b^k \end{cases}$$

Example 7

Solve $c_n \leq n^2 + c_{\lfloor \frac{n}{2} \rfloor} + c_{\lceil \frac{n}{2} \rceil}$

If c_n *is nondecreasing then* $c_{\lfloor \frac{n}{2} \rfloor} \leq c_{\lceil \frac{n}{2} \rceil}$

and we have $c_n \leq n^2 + 2c_{\lceil \frac{n}{2} \rceil}$

We have $a = 2, b = 2, f(n) = n^2$ and $k = 2.$

Since $a < b^k$, $c_n = O(n^k)$ *and since* $k = 2$, $c_n = O(n^2)$

If $T(n) \leq aT(\frac{n}{b}) + f(n)$ and $f(n) = O(n^k)$ then

$$T(n) = \begin{cases} O(n^k) & \text{if } a < b^k \\ O(n^k \log n) & \text{if } a = b^k \\ O(n^{\log_b a}) & \text{if } a > b^k \end{cases}$$

Example 8

Solve the following recurrences:

1. $T(n) = T(n-1) + n^r$
2. $T(n) = 2\,T(n-1) + n^r$

(Note: The solution will depend on the value of the parameter r and the initial conditions, which both need to be specified)

Example 9

$$T(n) = \begin{cases} 0 & \text{if } n = 0 \\ 2 + 4T\left(\frac{n}{2}\right) & \text{otherwise} \end{cases}$$

Restrict n to be a power of 2: $n = 2^k$

Look at the expansion:

$\boldsymbol{n}$	$\boldsymbol{T(n)}$
2^0	2
2^1	$2 + 4 \cdot 2$
2^2	$2 + 4 \cdot 2 + 4^2 \cdot 2$

Example 9 cont.

Pattern seems to suggest:

$$T(n) = T(2^k) = \sum_{i=0}^{k} 2.4^i = 2\frac{(4^{k+1}-1)}{4-1} = \frac{8n^2-2}{3} = O(n^2)$$

Proof by mathematical induction:

- Base case: $k = 0, n = 2^0$

  $T(1) = \frac{8-2}{3} = 2$ so the base case holds.

- Inductive Assumption: True for k.

- Inductive Step - Proof that it is true for $k+1$:

  $$T(2^{k+1}) = 2 + 4T(2^k) = 2 + 4\left(\frac{8 \times 2^{2k} - 2}{3}\right) = \frac{6 + 8 \times 2^{2k+2} - 8}{3}$$

  $$= \frac{8 \times 2^{2k+2} - 2}{3}$$

Smoothness

- **We assumed n is a power of 2**
  - This *gives* a conditional bound
  - $T(n)$ is in O $(n^2 \mid n$ is a power of $2)$
  - Can we remove this condition ?

  - **We can use "*smoothness*"** (FoA p.89)
    - basically show that *well-behaved* functions which have bounds at certain points, have the same bounds at all other points

Smoothness

A function f is smooth if:

- f is non-decreasing over the range $[N, +\infty)$
  - "eventually non-decreasing"
- For every integer $b \geq 2$
  - $f(bn)$ is $O(f(n))$

Let $f(n)$ be smooth. Let $t(n)$ be eventually non-decreasing. Then $t(n)$ is $\Theta(f(n))$ for n a power of b implies $t(n)$ is $\Theta(f(n))$ for any n.

Formally, a function f defined on positive integers is smooth if for any positive integer $b \geq 2$, there are positive constants C and N, depending on b, such that

$$f(bn) \leq Cf(n) \text{ and } f(n) \leq f(n+1)$$

for all $n \geq N$.

Smoothness

Lemma 1.

If t is a nondecreasing function, f is a smooth function, and $t(n) = \Theta(f(n))$ for *n* a power of b, then $t(n) = \Theta(f(n))$.

Which functions are smooth?

- log and polynomial functions are smooth, e.g., $\log(n)$, $\log^2(n)$, $n^3 - n$, $n\log(n)$, *etc.*
- Super-polynomial functions are <u>not</u>, e.g., 2^n, $n!$, *etc.*
- Using this, we can use guess-and-check a bit more loosely: prove for easier cases, and use *smoothness* for the rest

Proof of the Main Recurrence Theorem

Let a, b and k be integers satisfying $a \geq 1,\ b \geq 2$ and $k \geq 0$.

In the following, $\frac{n}{b}$ denotes either $\left\lfloor \frac{n}{b} \right\rfloor$ or $\left\lceil \frac{n}{b} \right\rceil$.

In the case of the floor function the initial condition $T(0) = u$ is given; and in the case of the ceiling function, the initial condition $T(1) = u$ is given.

We first need to show for both floor and ceiling $T(n)$ is well defined for all n. We use mathematical induction to prove that (to be done in tutorials).

Proof of the Main Recurrence Theorem

Next, we need to show that

$$\sum_{i=0}^{m} x^{m-i} y^i = \frac{x^{m+1} - y^{m+1}}{x - y}, \qquad x \neq y$$

We use mathematical induction to show this.

<u>Base Case:</u> $m = 0$

$x^0 y^0 = \frac{x^1 - y^1}{x - y}$, that is, $1 = 1$

<u>Inductive Assumption:</u> $m = k$

$$\sum_{i=0}^{k} x^{k-i} y^i = \frac{\left(x^{k+1} - y^{k+1}\right)}{x - y}$$

Proof of the Main Recurrence Theorem

Inductive Step: $m = k + 1$

$$\sum_{i=0}^{k+1} x^{k+1-i}y^i = \frac{(x^{k+2} - y^{k+2})}{x-y}$$

$$\sum_{i=0}^{k+1} x^{k+1-i}y^i = \sum_{i=0}^{k} x^{k+1-i}y^i + x^0y^{k+1} = \sum_{i=0}^{k} xx^{k-i}y^i + y^{k+1} =$$

$$= \frac{x\left(x^{k+1} - y^{k+1}\right)}{x-y} + y^{k+1} = \frac{x^{k+2} - xy^{k+1} + xy^{k+1} - y^{k+2}}{x-y} =$$

$$= \frac{x^{k+2} - y^{k+2}}{x-y}$$

Proof of the Main Recurrence Theorem

We are now ready to formulate a lemma that we shall latter use to prove the Main Recurrence Theorem.

Lemma 2.

If $n > 1$ is a power of b and $a \neq b^k$ the solution of recurrence relation $T(n) = aT(\frac{n}{b}) + cn^k$ is $T(n) = C_1 n^{\log_b a} + C_2 n^k$ for some constants C_1 and C_2, where $C_2 > 0$ for $a < b^k$ and $C_2 < 0$ for $a > b^k$.

If $n > 1$ is a power of b and $a = b^k$ the solution of recurrence relation $T(n) = aT(\frac{n}{b}) + cn^k$ is $T(n) = C_3 n^k + C_4 n^k \log_b n$ for some constants C_3 and $C_4 > 0$.

Proof of the Main Recurrence Theorem

<u>Proof:</u>

Suppose that $n = b^m$.

Then $m = \log_b n$, $n^k = (b^m)^k = (b^k)^m$ and $a^m = (b^{\log_b a})^m = n^{\log_b a}$

Then we have:

$$T(n) = aT(\frac{n}{b}) + cn^k$$

$$\begin{aligned}
T(n) = T(b^m) &= aT(b^{m-1}) + c(b^k)^m \\
&= a[aT(b^{m-2}) + c(b^k)^{m-1}] + c(b^k)^m \\
&= a^2T(b^{m-2}) + c[a(b^k)^{m-1} + (b^k)^m] \\
&= a^2[aT(b^{m-3}) + c(b^k)^{m-2}] + c[a(b^k)^{m-1} + (b^k)^m] \\
&= a^3T(b^{m-3}) + c[a^2(b^k)^{m-2} + a(b^k)^{m-1} + (b^k)^m] \\
&\dots \\
&= a^mT(b^0) + c\sum_{i=1}^{m} a^{m-i}(b^k)^i
\end{aligned}$$

Proof of the Main Recurrence Theorem

Proof (cont'd):

Thus $T(n) = a^m T(b^0) + c \sum_{i=1}^{m} a^{m-i}(b^k)^i$

If $a \neq b^k$, using $\sum_{i=0}^{m} x^{m-i} y^i = \frac{x^{m+1} - y^{m+1}}{x-y}$, $x \neq y$ we get

$$T(n) = a^m T(1) + c\,(\frac{(b^k)^{m+1} - a^{m+1}}{b^k - a} - a^m)$$

$$= a^m T(1) + c\,\frac{(b^k)^{m+1}}{b^k - a} + c\left(\frac{-a^{m+1}}{b^k - a} - a^m\right)$$

$$= a^m T(1) + c\,\frac{(b^k)^{m+1}}{b^k - a} - \frac{cb^k a^m}{b^k - a} = (T(1) - \frac{cb^k}{b^k - a})a^m + \frac{cb^k}{b^k - a}(b^k)^m =$$

$$= (T(1) - \frac{cb^k}{b^k - a})n^{\log_b a} + \frac{cb^k}{b^k - a}(b^k)^m = C_1 n^{\log_b a} + C_2 (b^k)^m$$

where $C_1 = T(1) - \frac{cb^k}{b^k - a}$ and $C_2 = \frac{cb^k}{b^k - a}$.

Note that if $a < b^k$ then $C_2 > 0$ and if $a > b^k$ then $C_2 < 0.$

Proof of the Main Recurrence Theorem

Proof (cont'd):

Recall $T(n) = a^m T(b^0) + c\sum_{i=1}^{m} a^{m-i}(b^k)^i$

If $a = b^k$, and we get

$$T(n) = a^m T(1) + c\sum_{i=1}^{m} a^m$$

$$= a^m T(1) + cma^m$$

$$= C_3 n^k + C_4 n^k \log_b n$$

where $C_3 = T(1)$ and $C_4 = c > 0.$

Proof of the Main Recurrence Theorem

Lemma 2 proves the main Recurrence Theorem for case where n is a power of b. We then apply Lemma 1 ("smooth" lemma) to generalize the proof for any n.