6 Recurrences

$T(n)$

Algorithm 2 mergesort(list L)

$O(n)$	$n \leftarrow \text{size}(L)$
$O(1)$	**if** $n \leq 1$ **return** L
$O(n)$	$L_1 \leftarrow L[1 \cdots \lfloor \frac{n}{2} \rfloor]$
	$L_2 \leftarrow L[\lfloor \frac{n}{2} \rfloor + 1 \cdots n]$
$T(\lfloor \frac{n}{2} \rfloor)$	mergesort(L_1)
$T(\lceil \frac{n}{2} \rceil)$	mergesort(L_2)
$O(n)$	$L \leftarrow \text{merge}(L_1, L_2)$
$O(1)$	**return** L

6 Recurrences

Algorithm 2 mergesort(list L)

$n \leftarrow \mathrm{size}(L)$
if $n \le 1$ **return** L
$L_1 \leftarrow L[1 \cdots \lfloor \frac{n}{2} \rfloor]$
$L_2 \leftarrow L[\lfloor \frac{n}{2} \rfloor + 1 \cdots n]$
$\mathrm{mergesort}(L_1)$
$\mathrm{mergesort}(L_2)$
$L \leftarrow \mathrm{merge}(L_1, L_2)$
return L

This algorithm requires

$$T(n) = T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \mathcal{O}(n) \le 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \mathcal{O}(n)$$

comparisons when $n > 1$ and 0 comparisons when $n \le 1$.

Recurrences

How do we bring the expression for the number of comparisons ($\approx$ running time) into a closed form?

For this we need to solve the recurrence.

Recurrences

How do we bring the expression for the number of comparisons ($\approx$ running time) into a closed form?

For this we need to solve the recurrence.

Methods for Solving Recurrences

1. **Guessing+Induction**
   Guess the right solution and prove that it is correct via induction. It needs experience to make the right guess.
2. **Master Theorem**
   For a lot of recurrences that appear in the analysis of algorithms this theorem can be used to obtain tight asymptotic bounds. It does not provide exact solutions.
3. **Characteristic Polynomial**
   Linear homogenous recurrences can be solved via this method.

Methods for Solving Recurrences

4. **Generating Functions**
   A more general technique that allows to solve certain types of linear inhomogenous relations and also sometimes non-linear recurrence relations.
5. **Transformation of the Recurrence**
   Sometimes one can transform the given recurrence relations so that it e.g. becomes linear and can therefore be solved with one of the other techniques.

6.1 Guessing+Induction

$f(n) = \Theta(n) \Rightarrow \exists c: f(n) \le c \cdot n \quad \forall n$

First we need to get rid of the $\mathcal{O}$-notation in our recurrence:

$$T(n) \le \begin{cases} 2T(\lceil \frac{n}{2} \rceil) + cn & n \ge 2 \\ 0 & \text{otherwise} \end{cases}$$

Informal way:

6.1 Guessing+Induction

First we need to get rid of the $\mathcal{O}$-notation in our recurrence:

$$T(n) \leq \begin{cases} 2T(\lceil \frac{n}{2} \rceil) + cn & n \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

Informal way:

Assume that instead we have

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

6.1 Guessing+Induction

First we need to get rid of the $\mathcal{O}$-notation in our recurrence:

$$T(n) \le \begin{cases} 2T(\lceil \frac{n}{2} \rceil) + cn & n \ge 2 \\ 0 & \text{otherwise} \end{cases}$$

Informal way:

Assume that instead we have

$$T(n) \le \begin{cases} 2T(\frac{n}{2}) + cn & n \ge 2 \\ 0 & \text{otherwise} \end{cases}$$

One way of solving such a recurrence is to guess a solution, and check that it is correct by plugging it in.

6.1 Guessing+Induction

Suppose we guess $T(n) \leq dn \log n$ for a constant d.

6.1 Guessing+Induction

Suppose we guess $T(n) \leq dn \log n$ for a constant d. Then

$$T(n) \leq 2T\left(\frac{n}{2}\right) + cn$$

6.1 Guessing+Induction

Suppose we guess $T(n) \leq dn \log n$ for a constant d. Then

$$\begin{aligned} T(n) &\leq 2T\left(\frac{n}{2}\right) + cn \\ &\leq 2\left(d\frac{n}{2}\log\frac{n}{2}\right) + cn \end{aligned}$$

6.1 Guessing+Induction

Suppose we guess $T(n) \leq dn\log n$ for a constant d. Then

$$\begin{aligned} T(n) &\leq 2T\Big(\frac{n}{2}\Big) + cn \\ &\leq 2\Big(d\frac{n}{2}\log\frac{n}{2}\Big) + cn \\ &= dn(\log n - 1) + cn \end{aligned}$$

6.1 Guessing+Induction

Suppose we guess $T(n) \leq dn \log n$ for a constant d. Then

$$\begin{aligned} T(n) &\leq 2T\left(\frac{n}{2}\right) + cn \\ &\leq 2\left(d\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c-d)n \end{aligned}$$

6.1 Guessing+Induction

Suppose we guess $T(n) \leq dn \log n$ for a constant d. Then

$$\begin{aligned} T(n) &\leq 2T\left(\frac{n}{2}\right) + cn \\ &\leq 2\left(d\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c-d)n \\ &\leq dn\log n \end{aligned}$$

if we choose $d \geq c$.

6.1 Guessing+Induction

Suppose we guess $T(n) \leq dn \log n$ for a constant d. Then

$$\begin{aligned} T(n) &\leq 2T\left(\frac{n}{2}\right) + cn \\ &\leq 2\left(d\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c-d)n \\ &\leq dn \log n \end{aligned}$$

if we choose $d \geq c$.

Formally, this is not correct if n is not a power of 2. Also even in this case one would need to do an induction proof.

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

- **base case** $(2 \leq n < 16)$:

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

- **base case** $(2 \leq n < 16)$: true if we choose $d \geq b$.

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

- **base case** $(2 \leq n < 16)$: true if we choose $d \geq b$.
- **induction step** $n/2 \to n$:

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

- **base case** $(2 \leq n < 16)$: true if we choose $d \geq b$.
- **induction step** $n/2 \to n$:

  Let $n = 2^k \geq 16$. Suppose statem. is true for $n' = n/2$. We prove it for n:

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

- **base case** $(2 \leq n < 16)$: true if we choose $d \geq b$.
- **induction step** $n/2 \to n$:

  Let $n = 2^k \geq 16$. Suppose statem. is true for $n' = n/2$. We prove it for n:

$$T(n) \leq 2T\left(\frac{n}{2}\right) + cn$$

6.1 Guessing+Induction

$$T(n) \le \begin{cases} 2T(\frac{n}{2}) + cn & n \ge 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \le dn\log n$.

Proof. (by induction)

- **base case** $(2 \le n < 16)$: true if we choose $d \ge b$.
- **induction step** $n/2 \to n$:

  Let $n = 2^k \ge 16$. Suppose statem. is true for $n' = n/2$. We prove it for n:

$$\begin{aligned} T(n) &\le 2T\Big(\frac{n}{2}\Big) + cn \\ &\le 2\Big(d\frac{n}{2}\log\frac{n}{2}\Big) + cn \end{aligned}$$

6.1 Guessing+Induction

$$T(n) \le \begin{cases} 2T(\frac{n}{2}) + cn & n \ge 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \le dn \log n$.

Proof. (by induction)

- **base case** $(2 \le n < 16)$: true if we choose $d \ge b$.
- **induction step** $n/2 \to n$:

  Let $n = 2^k \ge 16$. Suppose statem. is true for $n' = n/2$. We prove it for n:

$$\begin{aligned} T(n) &\le 2T\Big(\frac{n}{2}\Big) + cn \\ &\le 2\Big(d\frac{n}{2}\log\frac{n}{2}\Big) + cn \\ &= dn(\log n - 1) + cn \end{aligned}$$

6.1 Guessing+Induction

$$T(n) \le \begin{cases} 2T(\frac{n}{2}) + cn & n \ge 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \le dn\log n$.

Proof. (by induction)

- **base case** $(2 \le n < 16)$: true if we choose $d \ge b$.
- **induction step** $n/2 \to n$:

  Let $n = 2^k \ge 16$. Suppose statem. is true for $n' = n/2$. We prove it for n:

$$\begin{aligned} T(n) &\le 2T\left(\frac{n}{2}\right) + cn \\ &\le 2\left(d\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c-d)n \end{aligned}$$

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

- **base case** $(2 \leq n < 16)$: true if we choose $d \geq b$.
- **induction step** $n/2 \to n$:

  Let $n = 2^k \geq 16$. Suppose statem. is true for $n' = n/2$. We prove it for n:

$$\begin{aligned} T(n) &\leq 2T\left(\frac{n}{2}\right) + cn \\ &\leq 2\left(d\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c - d)n \\ &\leq dn\log n \end{aligned}$$

6.1 Guessing+Induction

$$T(n) \leq \begin{cases} 2T(\frac{n}{2}) + cn & n \geq 16 \\ b & \text{otw.} \end{cases}$$

Guess: $T(n) \leq dn \log n$.

Proof. (by induction)

- **base case** $(2 \leq n < 16)$: true if we choose $d \geq b$.
- **induction step** $n/2 \to n$:

  Let $n = 2^k \geq 16$. Suppose statem. is true for $n' = n/2$. We prove it for n:

$$\begin{aligned} T(n) &\leq 2T\left(\frac{n}{2}\right) + cn \\ &\leq 2\left(d\frac{n}{2}\log\frac{n}{2}\right) + cn \\ &= dn(\log n - 1) + cn \\ &= dn\log n + (c - d)n \\ &\leq dn\log n \end{aligned}$$

  Hence, statement is true if we choose $d \geq c$.

6.1 Guessing+Induction

How do we get a result for all values of n?

6.1 Guessing+Induction

How do we get a result for all values of n?

We consider the following recurrence instead of the original one:

$$T(n) \leq \begin{cases} 2T(\lceil \frac{n}{2} \rceil) + cn & n \geq 16 \\ b & \text{otherwise} \end{cases}$$

6.1 Guessing+Induction

How do we get a result for all values of n?

We consider the following recurrence instead of the original one:

$$T(n) \leq \begin{cases} 2T(\lceil \frac{n}{2} \rceil) + cn & n \geq 16 \\ b & \text{otherwise} \end{cases}$$

Note that we can do this as for constant-sized inputs the running time is always some constant (b in the above case).

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$\{2, \ldots, n-1\} \to n$

$$T(n)$$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$$T(n) \leq 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn$$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$$\begin{aligned} T(n) &\leq 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\ &\leq 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn \end{aligned}$$

6.1 Guessing+Induction

We also make a guess of $T(n) \le dn\log n$ and get

$$
\begin{aligned}
T(n) &\le 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\
&\le 2\left(d\left\lceil \frac{n}{2} \right\rceil \log\left\lceil \frac{n}{2} \right\rceil\right) + cn
\end{aligned}
$$

$\left\lceil \frac{n}{2} \right\rceil \le \frac{n}{2} + 1$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$$\begin{aligned}T(n) &\leq 2T\left(\left\lceil\frac{n}{2}\right\rceil\right) + cn \\ &\leq 2\left(d\left\lceil\frac{n}{2}\right\rceil \log\left\lceil\frac{n}{2}\right\rceil\right) + cn \\ &\leq 2\left(d(n/2+1)\log(n/2+1)\right) + cn\end{aligned}$$

$\left\lceil\frac{n}{2}\right\rceil \leq \frac{n}{2} + 1$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn\log n$ and get

$$T(n) \leq 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn$$

$$\leq 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn$$

$\left\lceil \frac{n}{2} \right\rceil \leq \frac{n}{2} + 1$

$$\leq 2\big(d(n/2+1)\log(n/2+1)\big) + cn$$

$\frac{n}{2} + 1 \leq \frac{9}{16}n$

6.1 Guessing+Induction

We also make a guess of $T(n) \le dn\log n$ and get

$$
\begin{aligned}
T(n) &\le 2T\left(\left\lceil\frac{n}{2}\right\rceil\right) + cn \\
&\le 2\left(d\left\lceil\frac{n}{2}\right\rceil\log\left\lceil\frac{n}{2}\right\rceil\right) + cn \\
\boxed{\left\lceil\tfrac{n}{2}\right\rceil \le \tfrac{n}{2} + 1} \quad &\le 2\big(d(n/2+1)\log(n/2+1)\big) + cn \\
\boxed{\tfrac{n}{2} + 1 \le \tfrac{9}{16}n} \quad &\le dn\log\left(\frac{9}{16}n\right) + 2d\log n + cn
\end{aligned}
$$

6.1 Guessing+Induction

We also make a guess of $T(n) \le dn\log n$ and get

$$
\begin{aligned}
T(n) &\le 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\
&\le 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn \\
&\le 2\left(d(n/2+1)\log(n/2+1)\right) + cn \\
&\le dn\log\left(\frac{9}{16}n\right) + 2d\log n + cn
\end{aligned}
$$

$\left\lceil \frac{n}{2} \right\rceil \le \frac{n}{2} + 1$

$\frac{n}{2} + 1 \le \frac{9}{16}n$

$\log \frac{9}{16}n = \log n + (\log 9 - 4)$

6.1 Guessing+Induction

We also make a guess of $T(n) \le dn \log n$ and get

$$
\begin{aligned}
T(n) &\le 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\
&\le 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn \\
\boxed{\left\lceil \tfrac{n}{2} \right\rceil \le \tfrac{n}{2} + 1} \quad &\le 2\left(d(n/2+1)\log(n/2+1)\right) + cn \\
\boxed{\tfrac{n}{2} + 1 \le \tfrac{9}{16}n} \quad &\le dn \log\left(\frac{9}{16}n\right) + 2d\log n + cn \\
\boxed{\log \tfrac{9}{16}n = \log n + (\log 9 - 4)} \quad &= dn \log n + (\log 9 - 4)dn + 2d\log n + cn
\end{aligned}
$$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$$\begin{aligned} T(n) &\leq 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\ &\leq 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn \\ \boxed{\left\lceil \tfrac{n}{2} \right\rceil \leq \tfrac{n}{2} + 1} \quad &\leq 2(d(n/2+1)\log(n/2+1)) + cn \\ \boxed{\tfrac{n}{2} + 1 \leq \tfrac{9}{16}n} \quad &\leq dn \log\left(\frac{9}{16}n\right) + 2d \log n + cn \\ \boxed{\log \tfrac{9}{16}n = \log n + (\log 9 - 4)} \quad &= dn \log n + (\log 9 - 4)dn + 2d \log n + cn \\ \boxed{\log n \leq \tfrac{n}{4}} \quad & \end{aligned}$$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$$
\begin{aligned}
T(n) &\leq 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\
&\leq 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn \\
\boxed{\left\lceil \tfrac{n}{2} \right\rceil \leq \tfrac{n}{2} + 1} \quad &\leq 2(d(n/2 + 1) \log(n/2 + 1)) + cn \\
\boxed{\tfrac{n}{2} + 1 \leq \tfrac{9}{16} n} \quad &\leq dn \log\left(\frac{9}{16} n\right) + 2d \log n + cn \\
\boxed{\log \tfrac{9}{16} n = \log n + (\log 9 - 4)} \quad &= dn \log n + (\log 9 - 4)dn + 2d \log n + cn \\
\boxed{\log n \leq \tfrac{n}{4}} \quad &\leq dn \log n + (\log 9 - 3.5)dn + cn
\end{aligned}
$$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$$
\begin{aligned}
T(n) &\leq 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\
&\leq 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn \\
\boxed{\left\lceil \tfrac{n}{2} \right\rceil \leq \tfrac{n}{2} + 1} \quad &\leq 2\big(d(n/2 + 1)\log(n/2 + 1)\big) + cn \\
\boxed{\tfrac{n}{2} + 1 \leq \tfrac{9}{16}n} \quad &\leq dn \log\left(\frac{9}{16}n\right) + 2d \log n + cn \\
\boxed{\log \tfrac{9}{16}n = \log n + (\log 9 - 4)} \quad &= dn \log n + (\log 9 - 4)dn + 2d \log n + cn \\
\boxed{\log n \leq \tfrac{n}{4}} \quad &\leq dn \log n + (\log 9 - 3.5)dn + cn \\
&\leq dn \log n - 0.33dn + cn
\end{aligned}
$$

6.1 Guessing+Induction

We also make a guess of $T(n) \leq dn \log n$ and get

$$
\begin{aligned}
T(n) &\leq 2T\left(\left\lceil \frac{n}{2} \right\rceil\right) + cn \\
&\leq 2\left(d\left\lceil \frac{n}{2} \right\rceil \log \left\lceil \frac{n}{2} \right\rceil\right) + cn \\
\boxed{\left\lceil \tfrac{n}{2} \right\rceil \leq \tfrac{n}{2} + 1} \quad &\leq 2\left(d(n/2+1)\log(n/2+1)\right) + cn \\
\boxed{\tfrac{n}{2} + 1 \leq \tfrac{9}{16}n} \quad &\leq dn \log\left(\frac{9}{16}n\right) + 2d\log n + cn \\
\boxed{\log \tfrac{9}{16}n = \log n + (\log 9 - 4)} \quad &= dn\log n + (\log 9 - 4)dn + 2d\log n + cn \\
\boxed{\log n \leq \tfrac{n}{4}} \quad &\leq dn\log n + (\log 9 - 3.5)dn + cn \\
&\leq dn\log n - 0.33dn + cn \\
&\leq dn\log n
\end{aligned}
$$

for a suitable choice of d.

6.2 Master Theorem

Lemma 5

Let $a \geq 1, b \geq 1$ *and* $\epsilon > 0$ *denote constants. Consider the recurrence*

$$T(n) = aT\left(\frac{n}{b}\right) + f(n) \;.$$

$\leftrightarrow n^{\log_b a}$

Case 1.

If $f(n) = \mathcal{O}(n^{\log_b(a)-\epsilon})$ *then* $T(n) = \Theta(n^{\log_b a})$.

Case 2.

If $f(n) = \Theta(n^{\log_b(a)} \log^k n)$ *then* $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$, $k \geq 0$.

Case 3.

If $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ *and for sufficiently large* n $af(\frac{n}{b}) \leq cf(n)$ *for some constant* $c < 1$ *then* $T(n) = \Theta(f(n))$.

6.2 Master Theorem

We prove the Master Theorem for the case that n is of the form b^ℓ, and we assume that the non-recursive case occurs for problem size 1 and incurs cost 1.

The Recursion Tree

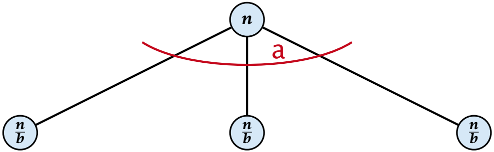
The running time of a recursive algorithm can be visualized by a recursion tree:

The Recursion Tree

The running time of a recursive algorithm can be visualized by a recursion tree:

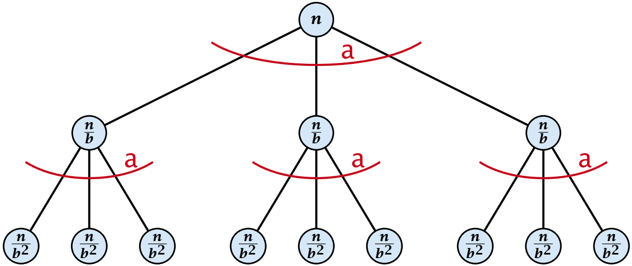
The Recursion Tree

The running time of a recursive algorithm can be visualized by a recursion tree:



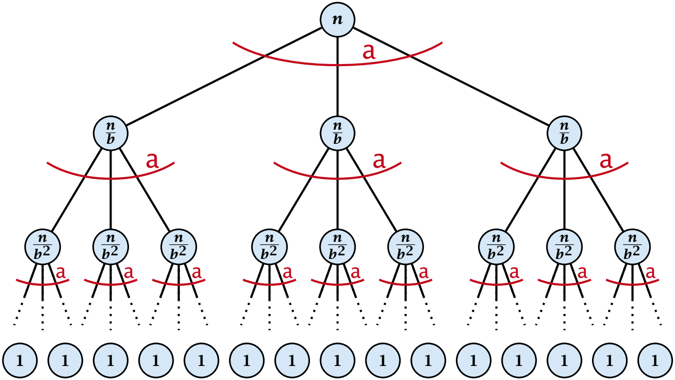
The Recursion Tree

The running time of a recursive algorithm can be visualized by a recursion tree:



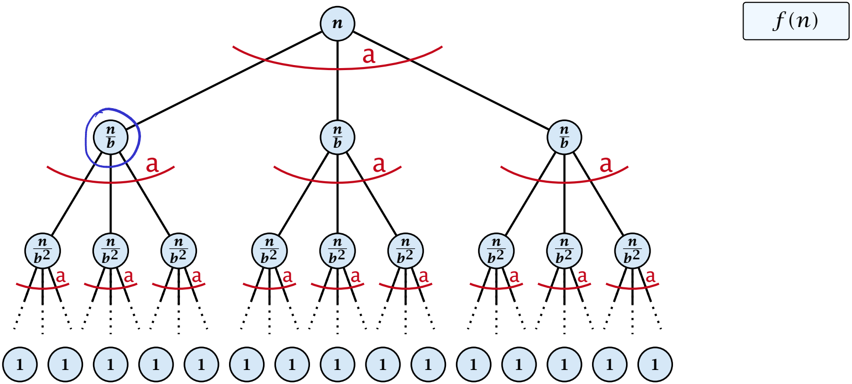
The Recursion Tree

The running time of a recursive algorithm can be visualized by a recursion tree:



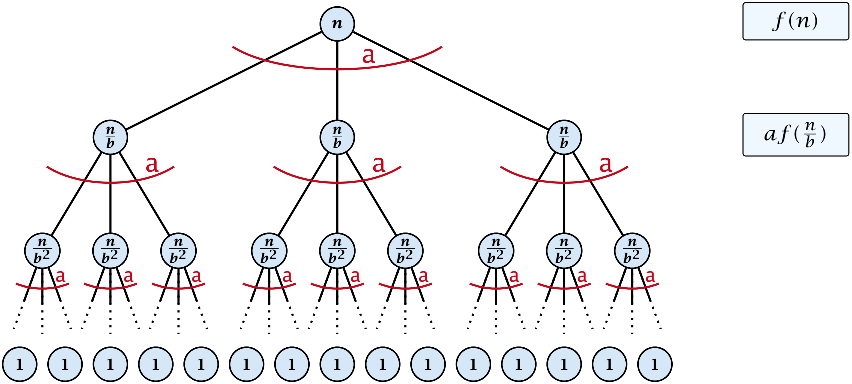
The Recursion Tree

The running time of a recursive algorithm can be visualized by a recursion tree:



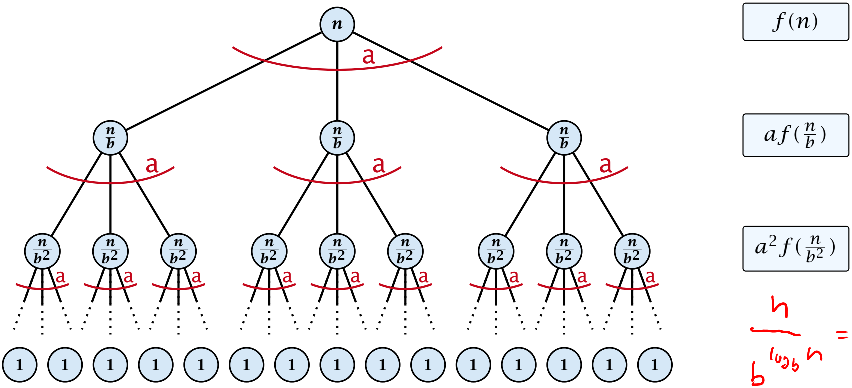
The Recursion Tree

The running time of a recursive algorithm can be visualized by a recursion tree:



The Recursion Tree

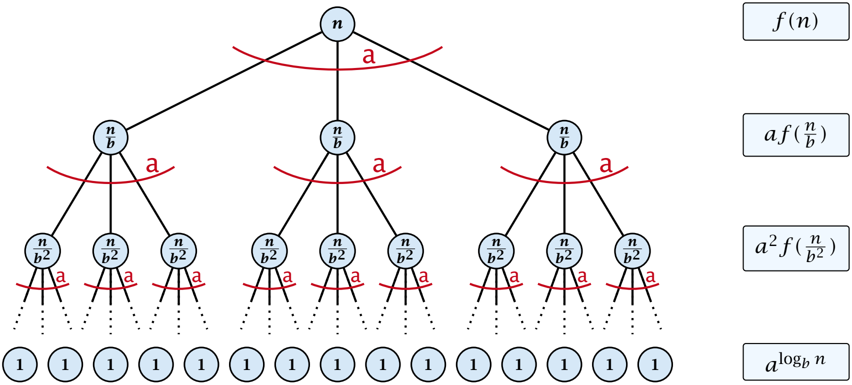
The running time of a recursive algorithm can be visualized by a recursion tree:



$$\frac{n}{b^{\log_b n}} = 1$$

The Recursion Tree

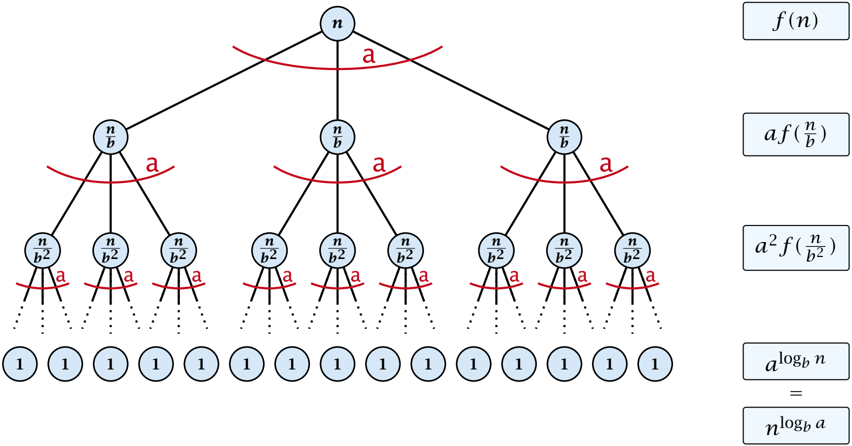
The running time of a recursive algorithm can be visualized by a recursion tree:



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

The Recursion Tree

The running time of a recursive algorithm can be visualized by a recursion tree:



6.2 Master Theorem

This gives

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

Case 1. Now suppose that $f(n) \leq cn^{\log_b a - \epsilon}$.

Case 1. Now suppose that $f(n) \leq cn^{\log_b a - \epsilon}$.

$$T(n) - n^{\log_b a}$$

Case 1. Now suppose that $f(n) \leq cn^{\log_b a-\epsilon}$.

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

Case 1. Now suppose that $f(n) \leq cn^{\log_b a - \epsilon}$.

$$
\begin{aligned}
T(n) - n^{\log_b a} &= \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right) \\
&\leq c \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a - \epsilon}
\end{aligned}
$$

Case 1. Now suppose that $f(n) \leq cn^{\log_b a - \epsilon}$.

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

$$\leq c \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a - \epsilon}$$

$$b^{-i(\log_b a - \epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i} a^{-i}$$

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

Case 1. Now suppose that $f(n) \leq cn^{\log_b a-\epsilon}$.

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

$$\leq c \sum_{i=0}^{\log_b n-1} a^i \left(\frac{n}{b^i}\right)^{\log_b a-\epsilon}$$

$$b^{-i(\log_b a-\epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i}a^{-i}$$

$$= cn^{\log_b a-\epsilon} \sum_{i=0}^{\log_b n-1} (b^{\epsilon})^i$$

$$\|$$

$$\sum_{i=0}^{\ell} q^i = \frac{q^{\ell+1}-1}{q-1}$$

Case 1. Now suppose that $f(n) \le cn^{\log_b a - \epsilon}$.

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

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

$$b^{-i(\log_b a - \epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i} a^{-i}$$

$$= cn^{\log_b a - \epsilon} \sum_{i=0}^{\log_b n - 1} (b^{\epsilon})^i$$

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

Case 1. Now suppose that $f(n) \le cn^{\log_b a - \epsilon}$.

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

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

$b^{-i(\log_b a - \epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i} a^{-i}$

$$= cn^{\log_b a - \epsilon} \sum_{i=0}^{\log_b n - 1} (b^{\epsilon})^i$$

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

$$= cn^{\log_b a - \epsilon}(b^{\epsilon \log_b n} - 1)/(b^{\epsilon} - 1)$$

Case 1. Now suppose that $f(n) \leq cn^{\log_b a - \epsilon}$.

$$
\begin{aligned}
T(n) - n^{\log_b a} &= \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right) \\
&\leq c \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a - \epsilon} \\
&= cn^{\log_b a - \epsilon} \sum_{i=0}^{\log_b n - 1} \left(b^{\epsilon}\right)^i \\
&= cn^{\log_b a - \epsilon}\left(b^{\epsilon \log_b n} - 1\right)/\left(b^{\epsilon} - 1\right) \\
&= cn^{\log_b a - \epsilon}\left(n^{\epsilon} - 1\right)/\left(b^{\epsilon} - 1\right)
\end{aligned}
$$

$b^{-i(\log_b a - \epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i} a^{-i}$

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

Case 1. Now suppose that $f(n) \leq cn^{\log_b a - \epsilon}$.

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

$$\leq c \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a - \epsilon}$$

$b^{-i(\log_b a - \epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i} a^{-i}$

$$= cn^{\log_b a - \epsilon} \sum_{i=0}^{\log_b n - 1} \left(b^{\epsilon}\right)^i$$

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

$$= cn^{\log_b a - \epsilon}(b^{\epsilon \log_b n} - 1)/(b^{\epsilon} - 1)$$

$$= cn^{\log_b a - \epsilon}(n^{\epsilon} - 1)/(b^{\epsilon} - 1)$$

$$= \frac{c}{b^{\epsilon} - 1} n^{\log_b a}(n^{\epsilon} - 1)/(n^{\epsilon})$$

≤ 1

Case 1. Now suppose that $f(n) \leq cn^{\log_b a-\epsilon}$.

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

$$\leq c \sum_{i=0}^{\log_b n-1} a^i \left(\frac{n}{b^i}\right)^{\log_b a-\epsilon}$$

$b^{-i(\log_b a-\epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i}a^{-i}$

$$= cn^{\log_b a-\epsilon} \sum_{i=0}^{\log_b n-1} (b^\epsilon)^i$$

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

$$= cn^{\log_b a-\epsilon}(b^{\epsilon \log_b n} - 1)/(b^\epsilon - 1)$$

$$= cn^{\log_b a-\epsilon}(n^\epsilon - 1)/(b^\epsilon - 1)$$

$$= \frac{c}{b^\epsilon - 1} n^{\log_b a}(n^\epsilon - 1)/(n^\epsilon)$$

Hence,

$$T(n) \leq \left(\frac{c}{b^\epsilon - 1} + 1\right) n^{\log_b(a)}$$

Case 1. Now suppose that $f(n) \leq cn^{\log_b a - \epsilon}$.

$$
\begin{aligned}
T(n) - n^{\log_b a} &= \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right) \\
&\leq c \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a - \epsilon}
\end{aligned}
$$

$b^{-i(\log_b a - \epsilon)} = b^{\epsilon i}(b^{\log_b a})^{-i} = b^{\epsilon i} a^{-i}$

$$
= cn^{\log_b a - \epsilon} \sum_{i=0}^{\log_b n - 1} (b^\epsilon)^i
$$

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

$$
\begin{aligned}
&= cn^{\log_b a - \epsilon}(b^{\epsilon \log_b n} - 1)/(b^\epsilon - 1) \\
&= cn^{\log_b a - \epsilon}(n^\epsilon - 1)/(b^\epsilon - 1) \\
&= \frac{c}{b^\epsilon - 1} n^{\log_b a}(n^\epsilon - 1)/(n^\epsilon)
\end{aligned}
$$

Hence,

$$
T(n) \leq \left(\frac{c}{b^\epsilon - 1} + 1\right) n^{\log_b (a)}
$$

$\Rightarrow T(n) = \mathcal{O}(n^{\log_b a})$.