Substitution Method

In the *substitution method*, we **guess** a bound and then use **mathematical induction** to prove our guess correct.

Q1: Determine an upper bound on the recurrence

$$T(n) = \begin{cases} 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Guess: We guess that the solution is $T(n) = O(n \lg n)$.

Proof by Induction: We have to prove that $T(n) \leq cn \lg n$ for an appropriate choice of the constant $c > 0$. We start by assuming that this bound holds for all positive $m < n$, in particular for $m = \left\lfloor \frac{n}{2} \right\rfloor$, yielding $T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \leq c \left\lfloor \frac{n}{2} \right\rfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right)$. ***Substituting*** into the recurrence yields

$$\begin{aligned} T(n) &\leq 2(c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \\ &\leq cn \lg(n/2) + n \\ &= cn \lg n - cn \lg 2 + n \\ &= cn \lg n - cn + n \\ &\leq cn \lg n , \end{aligned}$$

where the last step holds as long as $c \geq 1$.

Basis Step: As **log 1 = 0**, we make two new base cases, namely for **n = 2** & **n = 3**

- T(2) = 2 T(1) + 2 = 4 **<= 10 (2 lg 2)**
- T(3) = 2 T(1) + 3 = 5 **<= 10 (3 lg 3)**

Q2: Determine an upper bound on the recurrence

$$T(n) = \begin{cases} 3T\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + cn^2 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Guess: We guess that the solution is $T(n) = O(n^2)$ [Go through the **handout** on *Recursion Tree Method* to find the basis for this guess].

Proof by Induction: We have to prove that $T(n) \leq dn^2$ for an appropriate choice of the constant $d > 0$. We start by assuming that this bound holds for all positive $m < n$, in particular for $m = \frac{n}{4}$, yielding $T\left(\frac{n}{4}\right) \leq d\left(\frac{n}{4}\right)^2$. ***Substituting*** into the recurrence yields

$$\begin{aligned}
T(n) &\le 3T(\lfloor n/4 \rfloor) + cn^2 \\
&\le 3d\,\lfloor n/4 \rfloor^2 + cn^2 \\
&\le 3d(n/4)^2 + cn^2 \\
&= \frac{3}{16}dn^2 + cn^2 \\
&\le dn^2 ,
\end{aligned}$$

where the last step holds as long as $d \ge (16/13)c$.

Q3: Determine an upper bound on the recurrence

$$T(n) = \begin{cases} T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n) & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Guess: We guess that the solution is $T(n) = O(n \lg n)$ [Go through the **handout** on *Recursion Tree Method* to find the basis for this guess].

Proof by Induction: We have to prove that $T(n) \le dn\lg n$ for an appropriate choice of the constant $d > 0$. We start by assuming that this bound holds for all positive $m < n$, in particular for $m = \frac{n}{3}$ and $m = \frac{2n}{3}$, yielding $T\left(\frac{n}{3}\right) \le d\frac{n}{3}\lg\frac{n}{3}$ and $T\left(\frac{2n}{3}\right) \le d\frac{2n}{3}\lg\frac{2n}{3}$. ***Substituting*** into the recurrence yields

$$\begin{aligned}
T(n) &\le T(n/3) + T(2n/3) + cn \\
&\le d(n/3)\lg(n/3) + d(2n/3)\lg(2n/3) + cn \\
&= (d(n/3)\lg n - d(n/3)\lg 3) \\
&\qquad + (d(2n/3)\lg n - d(2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg 3 - (2n/3)\lg 2) + cn \\
&= dn\lg n - dn(\lg 3 - 2/3) + cn \\
&\le dn\lg n ,
\end{aligned}$$

as long as $d \ge c/(\lg 3 - (2/3))$.

Q4: What is wrong in the following proof to establish an upper bound on the recurrence

$$T(n) = \begin{cases} 2T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Proof: We want to prove that $T(n) \leq cn$ for an appropriate choice of the constant $c > 0$. We start by assuming that this bound holds for all positive $m < n$, in particular for $m = \left\lfloor \frac{n}{2} \right\rfloor$, yielding $T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \leq c\left\lfloor \frac{n}{2} \right\rfloor$. ***Substituting*** into the recurrence yields

$$\begin{aligned} T(n) &\leq 2(c\lfloor n/2 \rfloor) + n \\ &\leq cn + n \\ &= O(n)\,, \qquad \Longleftarrow \textit{wrong!!} \end{aligned}$$

The error is that we have not proved the ***exact form*** of the inductive hypothesis, that is, that $T(n) \leq cn$. We **must** explicitly prove that $T(n) \leq cn$ when we want to show that $T(n) = O(n)$.

Q5: Prove that $O(n)$ is an upper bound on the recurrence

$$T(n) = \begin{cases} T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + 1 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Proof: We have to prove that $T(n) \leq cn$ for an appropriate choice of the constant $c > 0$. Substituting our claim in the recurrence, we obtain

$$\begin{aligned} T(n) &\leq c\lfloor n/2 \rfloor + c\lceil n/2 \rceil + 1 \\ &= cn + 1\,, \end{aligned}$$

Here, once again we have not proved the ***exact form*** of the inductive hypothesis, that is, that $T(n) \leq cn$. Whereas, we **must** explicitly prove that $T(n) \leq cn$ when we want to show that $T(n) = O(n)$.

Intuitively, our guess is nearly right: we are off only by the constant 1, a lower-order term. Nevertheless, mathematical induction does not work unless we prove the exact form of the inductive hypothesis. We overcome our difficulty by *subtracting* a lower-order term from our previous guess. Our new guess is $T(n) \leq cn - d$, where $d \geq 0$ is a constant. We now have

$$\begin{aligned} T(n) &\leq (c\lfloor n/2 \rfloor - d) + (c\lceil n/2 \rceil - d) + 1 \\ &= cn - 2d + 1 \\ &\leq cn - d\,, \end{aligned}$$

as long as $d \geq 1$.