Recurrence relations:

A <u>recurrence relation</u> is a way to define a function by an expression involving the same function. (defining a function <u>inductively</u>)

The Fibonacci numbers are the most famous recurrence relation:

$$F(n) = F(n-1) + F(n-2), \quad F(0) = 1, \ F(1) = 1$$

We should have enough information to obtain $F(n)$ inductively, i.e. the base case and the inductive case. E.g. In above $F(1)=1$, $F(2)=1$ are the base case and the rest the inductive case.

Note that in above to obtain $F(n)$ we need to compute $F(1), \ldots, F(n-1)$ and thus $O(n)$ time, it is much easier if we have an explicit (or closed-form) expression for $F(n)$. This is called solving the recurrence relation. e.g.

$$F(n) = \frac{\varphi^n - \psi^n}{\sqrt{5}} \quad \text{where} \quad \underset{(\text{phi})}{\varphi} = \frac{1+\sqrt{5}}{2} \simeq \underset{(\text{golden ratio})}{1.6180} \quad \text{and} \quad \underset{(\text{psi})}{\psi} = \frac{1-\sqrt{5}}{2} = -0.618033$$

/fail/ /sail/

Recurrence relations are used a lot in the analysis of algorithms and thus we learn how to solve them.

Intelligent Guesses:

Guessing works well for a wide class of recurrence relations (of course needs playing exercises in advance). But let's see some standard forms. Substitution and expanding is another technique used a lot.

*) $a_n = r a_{n-1}$ and $a_0 = k$: Then $a_1 = rk$, $a_2 = r^2 k$, $a_n = k r^n$

*) $a_n = A a_{n-1} + B a_{n-2}$, $a_0 = k$, $a_1 = h$: let's guess $a_n = r^n$ and ~~replace~~ substitute it in the formula.

$r^n = A r^{n-1} + B r^{n-2} \Rightarrow r^2 = Ar + B$. Solve r to obtain two roots r_1, r_2.

Now it is easy to see that $c r_1^n$ and $d r_2^n$ and more generally $c r_1^n + d r_2^n$ (linear combination) are also the solutions and indeed the most general solutions if $r_1 \neq r_2$, and $C r_1^n + D n r_1^n$ if $r_1 = r_2$. C and D can be chosen based on two given initial values a_0 and a_1, i.e., $C + D = k$ and $C r_1 + D r_2 = h$ (solve the equations to obtain C and D).

For example if you solve $C + D = 1$ and $C r_1 + D r_2 = 1$ for Fibonacci numbers above we obtain the formula above. The equation $r^2 = Ar + B$ is called the <u>charateristic equation</u> and the same technique can be used for $a_n = A_1 a_{n-1} + A_2 a_{n-2} + \cdots + A_k a_{n-k}$.

Again note that the proof would be by induction (see wikipedia for more formulas)

(Proof by expansion)

*) $a_n = a_{n-1}+n$, $a_0 = k$ then $a_n = a_{n-1}+n = a_{n-2}+n-1+n = a_0+1+2+\dots+n = k+\frac{n(n+1)}{2}$

(Proof by substitution)

*) $a_n = 2a_{\lfloor \frac{n}{2} \rfloor}+n$ and $a_2=2$ [sometime in the running times we have $a_n \le a_{\lfloor \frac{n}{2} \rfloor}+n$ but we make it equality since we care about upper bound (O notation)]

First, we drop $\lfloor\ \rfloor$ and thus $a_n = 2a_{\frac{n}{2}}+n$ (alternatively assume n is even)

let $n=2^k$ and thus $k=\lg n$: thus $a_{2^k}=2a_{2^{k-1}}+2^k$

let $b_k = a_{2^k}$ and thus $b_1 = a_2$: hence $b_k = 2b_{k-1}+2^k$ (substitution method)

Now $a_n = a_{2^k} = b_k = 2b_{k-1}+2^k \underset{b_{k-1}=2b_{k-2}+2^{k-1}}{=\!=\!=} 2(2b_{k-2}+2^{k-1})+2^k = 2^2 b_{k-2}+2^k+2^k$

$\underset{b_{k-2}=2b_{k-3}+2^{k-2}}{=\!=\!=} 2^2(2b_{k-3}+2^{k-2})+2.2^k = 2^3 b_{k-3}+3\,2^k = \dots = 2^{k-1}b_1+(k-1)2^{k-1} = 2^{k-1}a_2+(k-1)2^{k-1}$

$= n+(\lg n-1)\frac{n}{2}$

$\frac{n}{2}(\lg n+1)$

Thus $a_n = O(n\log n)$ since Now you can come back and prove inductively that $a_n \le cn\log n$ for some constant c.

Basis $2=a_2 \le c(2.1)=2c$ which is correct for $c \ge 1$

IH: $a_{\lfloor \frac{n}{2} \rfloor} \le c\lfloor \frac{n}{2} \rfloor \log \lfloor \frac{n}{2} \rfloor$, then $a_n = 2a_{\lfloor \frac{n}{2} \rfloor}+n \le 2c\lfloor \frac{n}{2} \rfloor \log \lfloor \frac{n}{2} \rfloor + n \le 2c\frac{n}{2}\log\frac{n}{2}+n$

$\le cn(\log n-1)+n = cn\log n - cn+n \le cn\log n$ for $c \ge 1$. So the induction works for any $c \ge 1$.

Indeed the formula above is the number of comparisons in sorting n numbers, called merge sort: we divide the sequence into two parts of equal size $(\lfloor \frac{n}{2} \rfloor, \lceil \frac{n}{2} \rceil)$, we sort them $(a_{\lfloor \frac{n}{2} \rfloor}, a_{\lceil \frac{n}{2} \rceil})$ and then merge them with n more comparisons. Thus we have the formula above.