- **Problem 7.9.** $G^0 = 0$. $G_1 = 1$ and $G_n = 7G_{n-1} - 12G_{n-2}$ for $n > 1$. Compute G_5. Show $G_n = 4^n - 3^n$ for $n \geq 0$.

n	0	1	2	3	4	5
A_n	0	1	7	37	175	781

  (i) prove the base case:

$$G(0) = 4^0 - 3^0 = 0$$

base case is true

  (ii) prove $G(n) = 4^n - 3^n$ for $G(n) = 7G(n-1) - 12G(n-2)$ for $n > 1$

$$4^n - 3^n = 7G(n-1) - 12G(n-2)$$

manipulate the RHS

$$\begin{aligned}
4^n - 3^n &= 7(4^{n-1} - 3^{n-1}) - 12(4^{n-2} - 3^{n-2}) \\
&= 7(\frac{4^n}{4} - \frac{3^n}{3}) - 12(\frac{4^n}{16} - \frac{3^n}{9}) \\
&= 7(\frac{4^n \cdot 3 - 3^n \cdot 4}{12}) - 12(\frac{4^n \cdot 9 - 3^n \cdot 16}{144}) \\
&= 7(\frac{4^n \cdot 3 - 3^n \cdot 4}{12}) - (\frac{4^n \cdot 9 - 3^n \cdot 16}{12}) \\
&= \frac{21(4^n) - 28(3^n) - 9(4^n) + 16(3^n)}{12} \\
&= \frac{12(4^n) - 12(3^n)}{12} \\
&= 4^n - 3^n
\end{aligned}$$

  We prove the statement is true for $n \geq 0$ by direct proof ■

- **Problem 7.12(c). (See Problem 7.28 for hints.)** Tinker to guess a formula for each recurrence and prove it. In each case $A_1 = 1$ and for $n > 1$:

  (c) $A_n = 10nA_{n-1}/(n-1) + n$

n	2	3	4	5
A_n	22	333	4444	55555

  i. formula found:

$$\frac{10^n - 1}{9}n$$

ii. prove the base case:

$$\begin{aligned} A(2) &= \frac{100-1}{9}(2) \\ &= \frac{99}{9}(2) = 22 \end{aligned}$$

base case proven

iii. prove using direct proof

$$10n\frac{A(n-1)}{n-1} + n = \frac{10^n - 1}{9}(n)$$

with with LHS

$$\begin{aligned} 10n\frac{A(n-1)}{n-1} + n &= 10n\frac{\frac{10^{n-1}-1}{9}\cancel{(n-1)}}{\cancel{n-1}} + n \\ &= \cancel{(10)}n(\frac{10^n - 10}{\cancel{10}}\frac{1}{9}) + n \\ &= n\frac{10^n - 10}{9} + n \\ &= n\frac{10^n - 10}{9} + \frac{9n}{9} \\ &= \frac{10^n n - 10n + 9n}{9} \\ &= \frac{10^n n - n}{9} \\ \frac{10^n - 1}{9}n &= \frac{10^n - 1}{9}n \end{aligned}$$

iv. we prove by direct proof that the statement is true for all $n > 1$ ■

- **Problem 7.13(a).** Analyze these very fast growing recursions. [Hint: Take logarithms.]

(a) $M_1 = 2$ and $M_n = aM_{n-1}^2$ for $n > 1$. Guess and prove a formula for M_n. Tinker, tinker.

n	2	3	4	5
A_n	a4	a16	a256	a65536

(i) formula found:

$$M(n) = 2^{2^{n-1}}$$

(ii) base case:

$$\begin{aligned} M(2) &= a(2^{2^1}) \\ &= a(2^2) \\ &= a4 \end{aligned}$$

base case proven

(iii) prove using direct proof

$$\begin{aligned} aM(n-1)^2 &= a2^{2^{n-1}} \\ M(n-1)^2 &= 2^{2^{n-1}} \text{ simplify} \\ \log_2(M(n-1)^2) &= \log_2(2^{2^{n-1}}) \text{ log both sides} \\ \log_2(M(n-1)^2) &= 2^{2^{n-1}} \end{aligned}$$

work with LHS

$$\begin{aligned} \log_2(M(n-1)^2) &= 2\log_2(M(n-1)) \\ &= 2\log_2(2^{2^{(n-1)-1}}) \\ &= 2(2^{n-2}) \\ &= 2^{n-1} \end{aligned}$$

(iv) we prove by direct proof that the statement is true for all $n > 1$ ■

- **Problem 7.19(d).** Recall the Fibonacci numbers: F_1, $F_2 = 1$; and, $F_n = F_{n-1} + F_{n-2}$ for $n > 2$

  (d) Prove that every third Fibonacci number, F_{3n}, is even

  (i) We have to prove that $F_{3n} = 2p$, for some $p \in \mathbb{N}$

  (ii) prove the base case:

$$F_{3n-1} = F_2 \text{ when } n = 1, \ \rightarrow F_2 = 1$$
$$F_{3n-2} = F_1 \text{ when } n = 1, \ \rightarrow F_1 = 1$$

  (iii) since the fibonacci sequence is a sum of the previous two terms, we can make the following assumptions:

  By the given formula $F_n = F_{n-1} + F_{n-2}$, we can calculate $F_{3n} = F_{3n-1} + F_{3n-2}$

  We know that both F_{3n-1} and F_{3n-2} are sums of even and odd numbers

$$\begin{aligned} F_{3n-1} &= 2k + (2j+1) \\ &= 2(k+j) + 1 \\ F_{3n-2} &= (2w+1) + 2i \\ &= 2(w+i) + 1 \end{aligned}$$

plugging back into the original function, we get:

$$\begin{aligned}F_{3n} &= [2(k+j)+1] + [2(w+i)+1] \\ &= 2(k+j) + 2(w+i) + 2 \\ &= 2(k+j+w+i+1)\end{aligned}$$

We prove that the statement is true for all $n > 2$ ■

- **Problem 7.42.** Give pseudocode for a recursive function that computes 3^{2^n} on input n.

Code example:

```
int f(int n):
    if n is 0 return 3
    else return f(n-1) squared
```

Mathematical function:

$$\begin{aligned}T_0 &= 3 \\ T_n &= (T_{n-1})^2\end{aligned}$$

(a) Prove that your function correctly computes 3^{2^n} for every $n \geq 0$.

n	0	1	2	3	4
T_n	3	9	81	6561	43046721

(i) prove the base case for $n = 1$

$$\begin{aligned}T(n) &= T(n-1)^2 \\ T(1) &= T(0)^2 \\ &= 9\end{aligned}$$

(ii) prove using a direct proof

$$\begin{aligned}T(n) &= 3^{2^n} \\ T(n) &= T(n-1)^2\end{aligned}$$

$$
\begin{aligned}
T(n-1)^2 &= 3^{2^n} \\
(3^{2^{n-1}})^2 &= 3^{2^n} \text{ log both sides} \\
\log 3((3^{2^{n-1}})^2) &= \log 3(3^{2^n}) \\
2\log 3(3^{2^{n-1}}) &= \log 3(3^{2^n}) \\
2(2^{n-1}) &= 2^n \\
\text{LHS: } 2(2^{n-1}) &= 2^{n-1+1} \\
&= 2^n
\end{aligned}
$$

(iii) we prove by a direct proof that our function computes 3^{2^n} for every $n \geq 0$ ■

(b) Obtain a recurrence for the runtime T_n. Guess and prove a formula for T_n.

(i) runtime T_n

- assume squaring a number is passed onto a function such as:

```
int square(int n) {
    return n * n;
}
// two steps in total
// (1) multiplication
// (2) return
```

- $T_0 = 2$, when n is $0 \rightarrow$ (test, return)
- $T_1 = 6$, when n is $1 \rightarrow$ (test, multiplication(2), set, and T_0)
- $T_2 = 10$, when n is $2 \rightarrow$ (test, multiplication(2), set, and T_1)
- $T_n = T_{n-1} + 4$ for $n \geq 2$

(ii) derived formula: $T_n = 4n + 2$

base case: $n = 1$

$$
\begin{aligned}
T(1) &= 4(1) + 2 \\
&= 6
\end{aligned}
$$

prove by direct proof

$$
\begin{aligned}
T(n) &= T(n-1) + 4 \\
T(n) &= 4n + 2
\end{aligned}
$$

Setting both equations equal, we get

$$
\begin{aligned}
T(n-1)+4 &= 4n+2 \\
\text{LHS} \rightarrow T(n-1)+4 &= 4(n-1+2+4) \\
&= 4n \cancel{-4} + 2 \cancel{+4} \\
&= 4n+2
\end{aligned}
$$

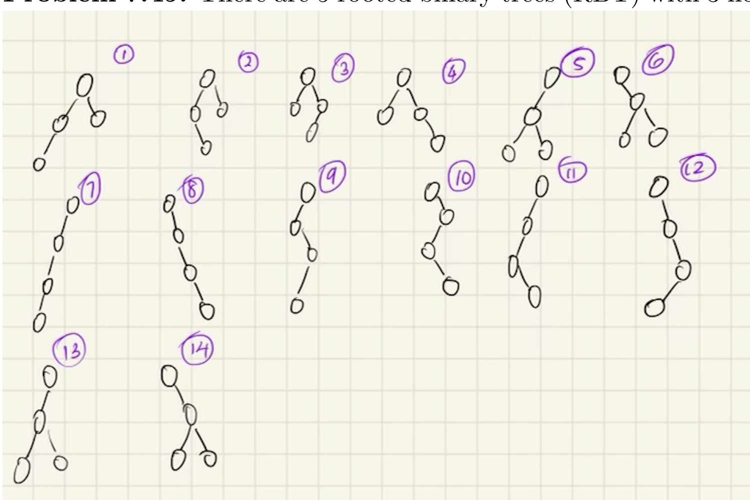
(iii) we prove by direct proof that our formula T_n accurately calculates the runtime T_n for $n \geq 1$ ■

- **Problem 7.45(c).** Give recursive definitions for the set S in each of the following cases.

  (c) $S = \{$all strings with the same number of 0's and 1's$\}$ (e.g. 0011,0101,100101).

  1. **[basis]** $\epsilon \in S, 0 \in S, 1 \in S$
  2. **[constructor(i)]** $\epsilon \in S \rightarrow 0 \bullet x \bullet 0 \in S$;
     [constructor(ii)] $\epsilon \in S \rightarrow 1 \bullet x \bullet 1 \in S$.

- **Problem 7.49.** There are 5 rooted binary trees (RBT) with 3 nodes. How many have 4 nodes



  - We can make 14 possible rooted binary trees with 4 nodes.

- **Problem 8.12(d).** A set P of parenthesis strings have a recursive definition.

  1. $\epsilon \in P$
  2. $x \in P \rightarrow [x] \in P$
     $x, y \in P \rightarrow xy \in P$

  (d) Prove by structural induction that every string P is balanced.

  i. hi

- **Problem 8.14.**