- **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) With induction hypothesis: assume $G(n) = 4^n - 3^n$, prove by direct proof that $G(n) = 4^n - 3^n \to G(n+1) = 4^{n+1} - 3^{n+1}$ for all $n \geq 0$.

$$\begin{aligned} G(n+1) &= 7G(n+1-1) - 12G(n+1-2) \text{ (recursive definition)} \\ &= 7G(n) - 12G(n-1) \\ 7G(n) - 12G(n-1) &= 4^{n+1} - 3^{n+1} \end{aligned}$$

manipulate the LHS

$$\begin{aligned} 7G(n) - 12G(n-1) &= 7(4^n - 3^n) - 12(4^{n-1} - 3^{n-1}) \text{ (induction hypothesis)} \\ &= 7(4^n - 3^n) - \cancel{12}(\frac{4^n(3) - 3^n(4)}{\cancel{12}}) \\ &= 7(4^n - 3^n) - (3(4^n) - 4(3^n)) \\ &= 7(4^n) - 7(3^n) - 3(4^n) - 4(3^n) \\ &= 4(4^n) - 3(3^n) \\ &= 4^{n+1} - 3^{n+1} \end{aligned}$$

  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:

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

base case proven

iii. With induction hypothesis: assume $A(n) = \frac{10^n-1}{9}n$, prove by direct proof that $A(n) = \frac{10^n-1}{9}n \to A(n+1) = \frac{10^{n+1}-1}{9}(n+1)$ for all $n > 0$.

$$A(n) = \frac{10nA(n-1)}{n-1} + n$$
$$A(n+1) = \frac{10(n+1)A(n)}{n} + (n+1) \text{ (recursive definition)}$$

with with RHS

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

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 $n > 0$ ■

- **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:

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

    (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 $0 \to$ (test, return)
- $T_1 = 6$, when n is $1 \to$ (test, multiplication(2), set, and T_0)
- $T_2 = 10$, when n is $2 \to$ (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

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

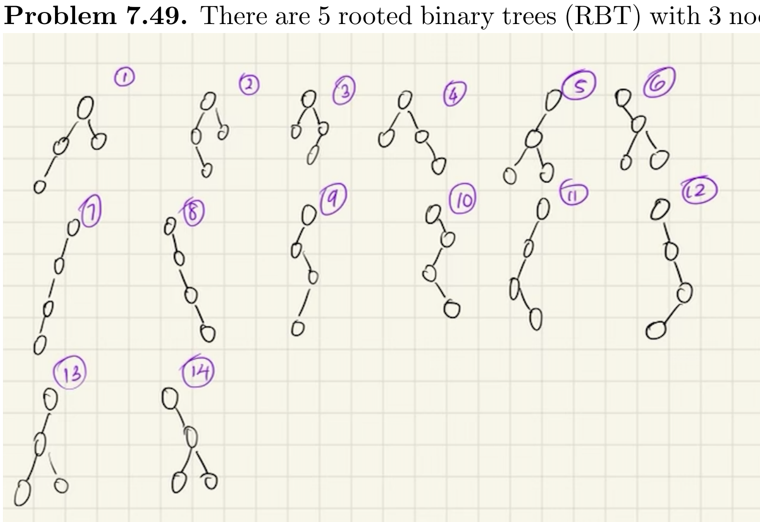
(iii) we prove by direct proof that our formula T_n accurately calculates the runtime T_n for $n \geq 1$ ■

- **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. **[Base case]** When $n = 1$ and $s_1 = \epsilon$, it is clearly balanced, $P(1)$ is true

  ii. **[Induction step]** We prove that each constructor preserves palindromicity.
  If x is a palindrome, that means x^R will be in P, or $x^R = x$. This is our induction hypothesis.

1. For constructor (i), we must show that $([x])^R = ([x])$.
   We can rewrite $[x]$ as $[\bullet x \bullet]$
   $([x])^R = [^R \bullet x^R \bullet]^R = [\bullet x^R \bullet] = [\bullet x \bullet] = [x]$
   A potential set of this could be:
   $P = \epsilon, [], [[]], [[[]]], ...,$ all preserving palindromicity.
2. For constructor(ii), we must show that $(xy)^R = xy$.
   We can rewrite xy as $x \bullet y$
   $(x \bullet y)^R = x^R \bullet y^R = x \bullet y = xy$
   A potential set of this could be: $P = \epsilon, [], [][], [][][][], ...,$ all preserving palindromicity.

iii. By structural induction, we prove that every string P is balanced given the constructors $\blacksquare$.

- **Problem 8.14.** A set A is defined recursively as shown.

1. $3 \in A$.
2. $x, y \in A \rightarrow x + y \in A$;
   $x, y \in A \rightarrow x - y \in A$.

(a) Prove that every element of A is a multiple of 3.

1. Prove by structural induction that every element in A is a multiple of 3.
2. **[Base case]** for P(0), we have both:

$$3 + 3 \in A = 6$$
$$3 - 3 \in A = 0$$

both are multiples of 3

3. **[Induction step]** suppose $x, y \in A$ and both x and y are multiples of 3

$$x = 3k$$
$$y = 3k$$

the constructor rules allow us to create the following formula:

$$x + y \in A$$
$$3k + 3w \in A$$
$$3(k + w) \in A$$

Adding two numbers that are multiples of 3 will always result in a number that is a multiple of 3

4. By structural induction, we conclude that ever member of A is a multiple of 3 $\blacksquare$

(b) Prove that every multiple of 3 is in A.

1. We prove by contradiction that every multiple of 3 is in A. Consider m, a multiple of 3 that is not in A.
2. **[Case 1]** $k > 0$, $m = 3k$, where m is the largest multiple of 3 NOT in our set A.

$$3k = 3 + 3 + 3 + ...$$

We can consider $3(k+1)$, which we know is in our set since it is larger than m

$$3(k+1) = 3k + 3$$

We know by constructor(ii) that $x - y \in A$, and we know that $3 \in A$ by the basis.

$$\begin{aligned} 3(k) &= x - y, \text{ where } x = 3k + 3 \text{ and } y = 3 \\ &= 3k + 3 - 3 \\ &= 3k \text{ woops, we derived a contradiction!} \end{aligned}$$

3. **[Case 2]** $k < 0$, $m = -3k$, where m is the smallest multiple of 3 NOT in our set A.

$$3(-k) = -3 - 3 - 3 - 3 - 3 - 3 - ...$$

We can consider $3(-k-1)$, which we know is in our set since it is smaller than m

$$3(-k-1) = -3k - 3$$

We know by constructor(i) that $x + y \in A$, and we know that $3 \in A$ by the basis.

$$\begin{aligned} -3k &= x + y, \text{ where } x = -3k - 3 \text{ and } y = 3 \\ &= -3k - 3 + 3 \\ &= -3k \text{ woops, we derived a contradiction!} \end{aligned}$$

4. **[Case 3]** $k = 0$, $m = 0$, 0 is not in A

We know from the basis that $x = 3$ and $y = 3$
From constructor(ii), we can use $x - y$, where:

$$\begin{aligned} x - y &\in A \\ 3 - 3 &\in A \\ 0 &\in A \text{ woops, we derived a contradiction!} \end{aligned}$$

5. We prove by contradiction for 3 distinct cases of k, proving that all multiples of 3 is in A ■