Math 15 - Chapters 3 and 4 Test

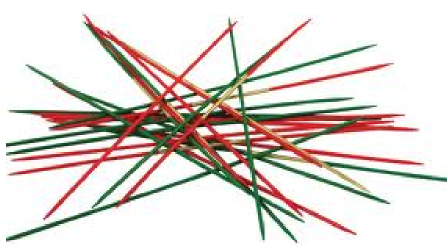
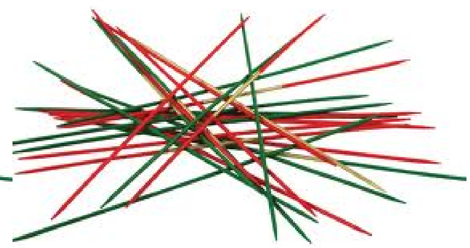
Show your work for each problem. Give thorough explanations in each case, using careful exposition to construct your answers. You may consult with others, but all the writing and understanding must be your own.

1. Let $C(n)$ be the coefficient of x^3 in the expansion of $(2x+3)^n$.
   Prove by induction on n that $C(n) = \binom{n}{3} 2^3 3^{n-3}$.

2. Let n be a positive integer. And let $P(n)$ be the statement that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using an el-shaped tile with three squares:

   Use induction to Prove $P(n)$. *Note: the removed square could be anywhere on the checkerboard.*

3. Two players take turns removing any positive number of matches they want from one of two piles of matches. The player who removes the last match wins the game.

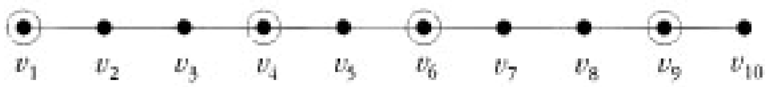
 

   Use induction to prove that if the two piles initially each contain the same number of sticks, the second player can always guarantee a win.

4. What is wrong with the following proof that every set of lines in the plane, no two of which are parallel, meet in a common point?
   Base case: $P(2)$ is true by the definition of parallel lines.
   Inductive hypothesis: Assume $P(k)$ is true, that is, every set of k lines meet in a common point.
   Inductive step: Consider a set of $k + 1$ line in the plane, no two of which are parallel. By the inductive hypotheses, the first k of them meet in a point, p. Also by the inductive hypothesis, the last k of these lines meet in a point q. If p and q were different points, then all the lines that contain both of them would be equal, a contradiction. Therefore, $p = q$ and all the lines meet at a single point.

5. Let F_n be the nth Fibonacci number and let L_n be the nth Lucas number. Prove the following:
   a. $F_{m+n} = F_{m+1}F_n + F_mF_{n-1}$
   b. $L_{m+n} = F_{m+1}L_n + F_mL_{n-1}$
   c. $F_{n-1} + F_{n+1} = L_n$
   d. $L_{n-1} + L_{n+1} = 5F_n$
   e. $F_nL_n = F_{2n}$
   f. $F_n^2 = F_{n-1}F_{n+1} + (-1)^{n-1}$

6. The following problems are related.
   a. Let a_n denote the number of length *n* binary sequence with no consecutive 0's. For instance, 0110101101. Show that a_n satisfies the recurrence relation, $a_n = a_{n-1} + a_{n-2}$.
   b. Let b_n denote the number of ways to select a subset of nonadjacent vertices from a path on *n* vertices (as in the figure below, where nonadjacent vertices v_1, v_4, v_6, v_9 are chosen.) Such a subset of vertices is called an *independent set*. Notice that b_{n-2} and b_{n-1} count the number of independent subsets that do and do not contain the first point on the path, respectively. Show that $b_n = F_{n+2}$, a Fibonacci number.

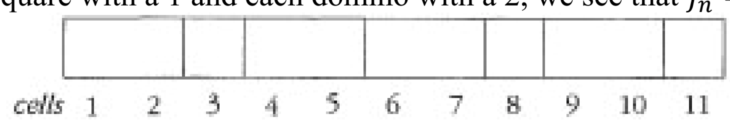


   This establishes a correspondence between independent sets of vertices and the 0's of binary sequences. Note that 0110101101 from part (a) corresponds to the independent set above.
   c. Note that there is a natural correspondence between these two representations, a_n and b_n: independent sets of vertices correspond to 0's in in the binary sequences.

   Let c_n denote the number of series of 1's and 2's that add to *n*. Then $c_1 = 1$ and $c_2 = 2$ since 1 = 1 is the only way to represent 1 and 2 = 2 = 1+1 are the two ways to represent 2. Prove that $c_n = F_{n+1}$.

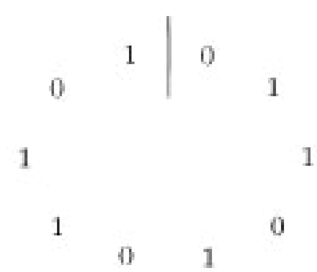
   Note that there is a natural correspondence between c_{n+1} and b_n. For a given series of 1's and 2's that add to *n* + 1, associate the subset of vertices whose *indices* are not partial sums of the series. For example, the series 2 + 1 + 2 + 2 + 1 + 2 + 1 = 11 has partial sums 2, 3, 5, 7, 8, 10 and 11 yielding the independent set v_1, v_4, v_6, v_9.

   d. Now consider the number of ways to tile a $1 \times n$ checkerboard with cells labeled $1, 2, \ldots, n$. Let f_n denote the number of ways to tile an *n*-board with 1×1 *squares* and 1×2 *dominoes*. Associating each square with a 1 and each domino with a 2, we see that $f_n = c_n$.



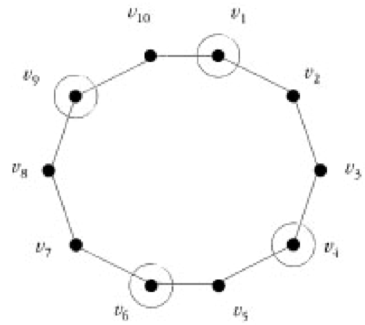
   Thus, $f_n = F_{n+1}$. So f_{m+n} is the number of ways to tile a length $m + n$ board. Explain why $f_{m+n} = f_m f_n + f_{m-1} f_{(n-1)}$.

7. Lucas numbers act like Fibonacci numbers running in circles.
   a. Let A_n denote the number of length *n circular* binary sequences with no consecutive 0's (as in the figure.) What are the length 2 and 3 circular sequences?

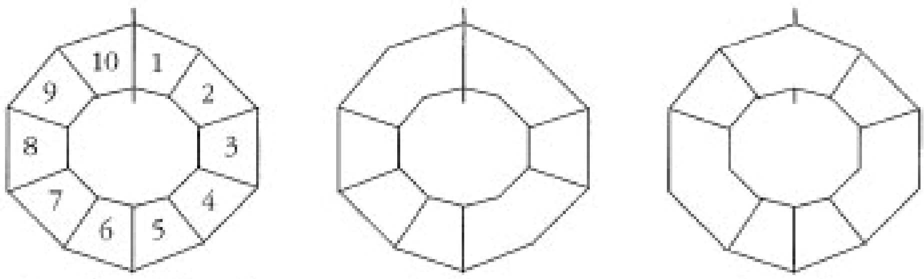


   b. Prove that $A_n = L_n$, the *n*th Lucas number.

c. Let $B_n = A_n = L_n$ denote the number of independent sets in a cycle graph with n vertices.
   Let C_n denote the number of series of 1's and 2's that sum to n with the end point restriction that it may not begin and end with a 2. Show that $C_n = L_{n-1}$.



d. Let l_n denote the number of ways to tile a circular $1 \times n$ board with squares and dominoes. Cells are labeled 1 through n and a tiling is called an n-bracelet. (see below.) An n-bracelet is *out of phase* if a domino



Circular 10-board — In phase — Out of phase

covers cells n and 1, otherwise the n-bracelet is *in phase*. Show that the number of in phase n-bracelets is $f_n = F_{n+1}$ and the number of out of phase n-bracelets is $f_{n-2} = F_{n-1}$ and that $l_n = L_n$.

8. Consider the following function *p*, where *L* is a list.
   B. If $L = x$, a single element, then $p(L)$ = "x".
   R. If $L = L', x$ for some list L', then $p(L)$ = "$x, p(L')$".
   If L = `john, paul, george, ringo,` what is $p(L)$?

9. Suppose *L* is an SList of depth *p*. Find a recurrence relation for $A(p)$, the number of times two numbers are added when evaluating Sum(L).

10. Let *L* be an SList. Define a recursive function Wham as follows.
    B. Suppose $L = x$. Then Wham(L) = $x \cdot x$.
    R. Suppose $L = (X, Y)$. Then
    Wham(L) = Wham(X) + Wham(Y).
    a. Evaluate Wham$\big((2,4)(6,7)\big)$. Remember to show all work.
    b. Give a recurrence relation for $S(p)$, the number of + operations performed by Wham on an SList of depth p, for $p \geq 0$.
    c. Give a recurrence relation for $M(p)$, the number of $\cdot$ operations performed by Wham on an SList of depth p, for $p \geq 0$.D

11. An urn contains six red balls, six white balls, and six blue balls, and sample of three balls is drawn at random without replacement. Compute the probability that the sample contains at least one ball of each color. (Round your answer to four decimal places.)

12. An urn contains two red balls and five blue balls. Draw two balls at random from the urn, without replacement. Compute the expected number of red balls in your sample. (Round your answer to four decimal places.)

13. Consider the following algorithm.

```
for i ∈ {1, 2, 3, 4} do
  ┌ beep
  │ For j ∈ {1, 2, 3} do
  │    beep
  │ for k ∈ {1, 2, 3, 4} do
  │    ┌ for l ∈ {1, 2, 3, 4, 5, 6} do
  │    │    beep
  │    │ for m ∈ {1, 2, 3, 4, 5} do
  └    └    beep
```

How many times does a **beep** statement get executed?

14. Let $x_1, x_2, \ldots, x_n$ be an array. Consider the following algorithm.

for $i \in \{1, 2, \ldots, \lfloor n/2 \rfloor\}$ **do**

┌ $t \leftarrow x_i$

$x_i \leftarrow x_{n-i+1}$

└ $x_{n-i+1} \leftarrow t$

a. How many ⟵ operations does this algorithm perform? Your answer should be a function of *n*.
b. What does this algorithm do to the array?

15. An urn contains *m* red balls and *n* green balls.

a. Give a big-Θ estimate for the number of ways to draw a sequence of *n* green balls without replacement.
b. Give a big-Θ estimate for the number of ways to choose 2 red balls and 3 green balls (assuming $m > 1$ and $n > 2$) without replacement.

Math 15 - Chapters 3 and 4 Test Solutions

1. Let $C(n)$ be the coefficient of x^3 in the expansion of $(2x+3)^n$.
   Prove by induction on n that $C(n) = \binom{n}{3}2^3 3^{n-3}$.
   SOLN: It would be helpful to have a few lemmas at our disposal here.
   Lemma 1: The constant term is 3^n. Proof, for $n = 1$ it works. Assume the constant term of $(2x+3)^{k-1}$ is 3^{k-1}, then then $(2x+3)^k = (2x+3)(2x+3)^{k-1}$ has the constant term $3 \cdot 3^{k-1} = 3^k$.
   Lemma 2: The coefficient of x in the expansion of $(2x+3)^n$ is $2n3^{n-1}$. Proof, for $n = 1$ it works. Assume the coefficient of x in $(2x+3)^{k-1}$ is $2n3^{k-2}$, then, using lemma1 and the inductive hypothesis,
   $(2x+3)^k = (2x+3)(2x+3)^{k-1} = (2x+3)\left(\cdots + 2(k-1)3^{k-2}x + 3^{k-1}\right)$
   $= \cdots + \left(2 \cdot 3^{k-1} + 2(k-1)3^{k-1}\right)x + 3^k = \cdots + \left(2+2(k-1)\right)3^{k-1}x + 3^k = \cdots + (2k)3^{k-1}x + 3^k$, so the coefficient of x is $2k3^{k-1}$, as desired.
   Lemma 3: The coefficient of x^2 in the expansion of $(2x+3)^n$ is $2n(n-1)3^{n-2}$.
   Proof: For $n = 2$, it works. Assume that the coefficient of x^2 in $(2x+3)^{k-1}$ is $2(k-1)(k-2)3^{k-3}$ then, using lemma 2 and this inductive hypothesis,
   $(2x+3)^k = (2x+3)(2x+3)^{k-1} = (2x+3)\left(\cdots + 2(k-1)(k-2)3^{k-3}x^2 + 2(k-1)3^{k-2}x + 3^{k-1}\right)$
   $= \cdots + 6(k-1)(k-2)3^{k-3}x^2 + 4(k-1)3^{k-2}x^2 + \cdots = \cdots + 2(k-1)3^{k-3}(3k-6+6)x^2 + \cdots$
   $= \cdots + 2k(k-1)3^{k-2}x^2 + \cdots$
   Now we can prove the claim: For the base case we use $n = 3$ and observe that the coefficient of x^3 in $(2x+3)^3 = 8x^3 + \cdots$ is $8 = \binom{3}{3}2^3 3^{3-3}$. Now assume that the coefficient of x^3 in $(2x+3)^{k-1}$ is $(k-1) = \binom{k-1}{3}2^{k-1}3^{k-4}$. Then the inductive step is $(2x+3)^k = (2x+3)(2x+3)^{k-1} =$
   $(2x+3)\left(\cdots + \binom{k-1}{3}2^3 3^{k-4}x^3 + 2(k-1)(k-2)3^{k-3}x^2 + 2(k-1)3^{k-2}x + 3^{k-1}\right) =$
   $\cdots + \binom{k-1}{3}2^3 3^{k-3}x^3 + 4(k-1)(k-2)3^{k-3}x^3 + \cdots$
   $= \cdots + \left(\frac{(k-1)(k-2)(k-3)}{6}2^3 3^{k-3} + 4(k-1)(k-2)3^{k-3}\right)x^3 + \cdots$
   $= \cdots + (k-1)(k-2)3^{k-4}\left((k-3)2^2 + 12\right)x^3 + \cdots$
   $= \cdots + 4k(k-1)(k-2)3^{k-4}x^3 + \cdots$
   $= + \cdots \binom{k}{3}2^3 3^{k-3}x^3 + \cdots$. QED.

   This may not have been the best proof…not the proof from "The Book." This is a special case of the binomial theorem, after all, and it might be easier just to prove the binomial theorem.

   For all $n \in \mathbb{N}$, $(a+b)^n = \sum_{i=0}^{n}\binom{n}{i}a^i b^{n-i}$.
   Proof. (Using mathematical induction)
   Let P_n be the statement $(a+b)^n = \sum_{i=0}^{n}\binom{n}{i}a^i b^{n-i}$
   <u>Basis step</u>. Show P_1 is true.
   For $n = 1$, the right side of P_1 is $\sum_{i=0}^{1}\binom{1}{i}a^i b^{1-i} = \binom{1}{0}a^0 b^{1-0} + \binom{1}{1}a^1 b^{1-1} = b + a = a + b$ and the left side of P_1 is $(a+b)^1$ So P_1 is true.

Induction hypothesis.

Let k be an integer and assume P_k is true, that is, assume that $(a\ +\ b)^k\ = \sum_{i=0}^{k} \binom{k}{i} a^i b^{k-i}$

Inductive step: $P_k \Rightarrow P_{k+1}$ is true, i.e. we $(a\ +\ b)^{k+1}\ = \sum_{i=0}^{k+1} \binom{k+1}{i} a^i b^{k+1-i}$

Consider $\sum_{i=0}^{k} \binom{k}{i} a^i b^{k-i}$ and recall the formula for Pascal's triangle: $\binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1}$ which is true for $1 \le i \le n-1$. Now,

$$(a+b)^{k+1} = (a+b)(a+b)^k = a(a+b)^k + b(a+b)^k = a\sum_{i=0}^{k}\binom{k}{i} a^i b^{k-i} + b\sum_{i=0}^{k}\binom{k}{i} a^i b^{k-i}$$

$$= a\left(a^k + \binom{k}{1} a^{k-1}b + \cdots + \binom{k}{i} a^{k-i}b^i + \cdots + \binom{k}{k-1} ab^{k-1} + b^k\right)$$

$$+b\left(a^k + \binom{k}{1} a^{k-1}b + \cdots + \binom{k}{i} a^{k-i}b^i + \cdots + \binom{k}{k-1} ab^{k-1} + b^k\right) \text{ (by inductive hypothesis.)}$$

$$= \left(a^{k+1} + \binom{k}{1} a^k b + \cdots + \binom{k}{i} a^{k+1-i}b^i + \cdots + \binom{k}{k-1} ab^{k-1} + ab^k\right)$$

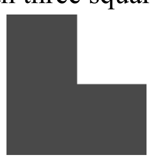
$$+ \left(ba^k + \binom{k}{1} a^{k-1}b^2 + \cdots + \binom{k}{i} a^{k-i}b^{i+1} + \cdots + \binom{k}{k-1} ab^k + b^{k+1}\right)$$

$$= a^{k+1} + \left[\left(\binom{k}{1} + \binom{k}{0}\right) a^k b + \left(\binom{k}{2} + \binom{k}{1}\right) a^{k-1}b^2 + \cdots + \left(\binom{k}{i} + \binom{k}{i-1}\right) a^{k+1-i}b^i + \cdots + \right.$$

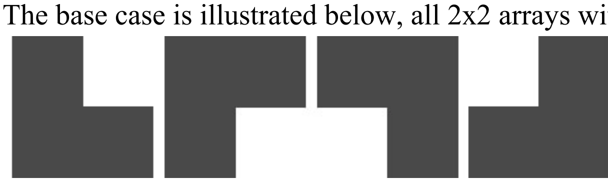
$$\left.\left(\binom{k}{k} + \binom{k}{k-1}\right) ab^k\right] + b^{k+1} = \sum_{i=0}^{k+1} \binom{k+1}{i} a^i b^{k+1-i} \text{ by Pascal's formula.}$$

Hence P_{k+1} is true if P_k is true. By the principle of mathematical induction, P_n is true for all $n \in N$, and the binomial theorem is proved.

2. Let n be a positive integer. And let $P(n)$ be the statement that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using an el-shaped tile with three squares:

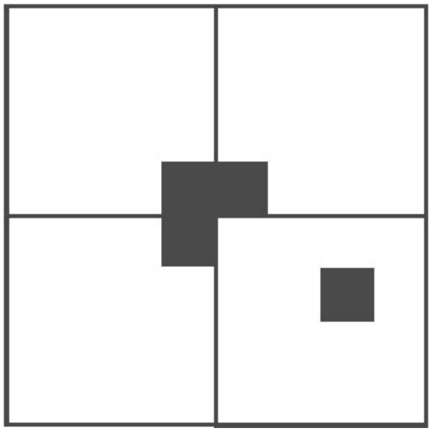


Use induction to Prove $P(n)$. *Note: the removed square could be anywhere on the checkerboard.*
SOLN: Claim: every $2^n \times 2^n$ checkerboard with one square removed can be tiled using an el-shaped tile with three squares.
The base case is illustrated below, all 2x2 arrays with on square removed:

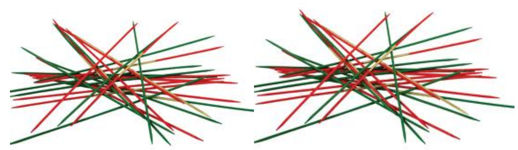


Inductive hypothesis: Suppose that every $2^k \times 2^k$ checkerboard with one square removed can be tiled with el-shaped tiles. To see that this implies a $2^{k+1} \times 2^{k+1}$ checkerboard can be tiled with el-shaped tiles, split the big checkerboard into 4 quarter-sized checkerboards. Put one el-shaped tile covering 3 corner squares of these quarter-sized pieces and in the center of the $2^{k+1} \times 2^{k+1}$ checkerboard as shown below. By the inductive hypothesis, the rest of those quarter sized pieces can be tiled. Now remove any arbitrary cell of the remaining quarter-sized checkerboard – the rest of it can also be tiled by hypothesis.

Rotating the checkerboard, we can place the arbitrary cell anywhere in the $2^{k+1} \times 2^{k+1}$ checkerboard, so the mathematical induction is complete.



3. Two players take turns removing any positive number of sticks they want from one of two piles of sticks. The player who removes the last stick wins the game.



Use induction to prove that if the two piles initially each contain the same number of sticks, the second player can always guarantee a win.
SOLN: Let $P(n)$ denote the statement, "the second player wins when there are initially n sticks in each pile.
Basis Step: $P(1)$ is true because in this case there is only one stick in each pile, and the first player has only one choice, removing one stick from the pile. Then the second player removes the stick from the other pile and wins.
Inductive step: Suppose that $P(j)$ is true for all j with $1 \leq j \leq k$. We need to show this implies that $P(k + 1)$ is true, that is, that the second player wins when each pile contain $k + 1$ matches. Suppose that the first player removes r matches from one pile leaving $k + 1 - r$ matches there. By removing the same number of matches from the other pile the second player creates the situation of two piles with $k + 1 - r$ matches in each. Apply the inductive hypothesis.

4. What is wrong with the following proof that every set of lines in the plane, no two of which are parallel, meet in a common point?
Base case: $P(2)$ is true by the definition of parallel lines.
Inductive hypothesis: Assume $P(k)$ is true, that is, every set of k lines meet in a common point.
Inductive step: Consider a set of $k + 1$ line in the plane, no two of which are parallel. By the inductive hypotheses, the first k of them meet in a point, p. Also by the inductive hypothesis, the last k of these lines meet in a point q. If p and q were different points, then all the lines that contain both of them would be equal, a contradiction. Therefore, $p = q$ and all the lines meet at a single point.
SOLN: The most obvious thing wrong with the "proof" is that it is easy to produce a counterexample: any 3 lines that produce a triangle are lines which are not concurrent and have no two lines parallel to one another. I like Joe Moeller's argument: "…the [proof] requires the intersection of the two subsets (first k and last k) must be non-empty, but in the base case, $k = 2$, the intersection is empty (a line does not intersect itself at a unique point), so it's not a valid base case." Note the implicit definition of "first k" and "last k" is unclear. For $k = 3$: if you try to break these 3 up into the "first 2" and the "last 2" it's unclear which 2 of the 3 pairs these would be.

5. Let F_n be the *n*th Fibonacci number and let L_n be the *n*th Lucas number. Prove the following:

a. $F_{m+n} = F_{m+1}F_n + F_mF_{n-1}$

Proof: Base case: If *n* = 2 and *m* = 1, then $F_3 = F_2F_2 + F_1F_1 = 1 + 1 = 2$ is true.

Note that $\begin{pmatrix}0 & 1\\1 & 1\end{pmatrix}^n = \begin{pmatrix}F_{n-1} & F_n\\F_n & F_{n+1}\end{pmatrix}$ so

$\begin{pmatrix}0 & 1\\1 & 1\end{pmatrix}^{m+n} = \begin{pmatrix}0 & 1\\1 & 1\end{pmatrix}^m \begin{pmatrix}0 & 1\\1 & 1\end{pmatrix}^n = \begin{pmatrix}F_{m-1} & F_m\\F_m & F_{m+1}\end{pmatrix}\begin{pmatrix}F_{n-1} & F_n\\F_n & F_{n+1}\end{pmatrix} =$

$\begin{pmatrix}F_{m-1}F_{n-1} + F_mF_n & F_{m-1}F_n + F_mF_{n+1}\\F_mF_{n-1} + F_{m+1}F_n & F_mF_n + F_{m+1}F_{n+1}\end{pmatrix} = \begin{pmatrix}F_{m+n-1} & F_{m+n}\\F_{m+n} & F_{m+n+1}\end{pmatrix}$

Looking at the second row, first column positions does the trick.

b. $L_{m+n} = F_{m+1}L_n + F_mL_{n-1}$

SOLN: Let *m* = 1. Then the equation is $L_{n+1} = F_2L_n + F_1L_{n-1} = L_n + L_{n-1}$ is the recurrence relation for the Lucas numbers, so that's a good basis step.

Inductive hypothesis: Assume that $L_{m+n} = F_{m+1}L_n + F_mL_{n-1}$ for all $1 \le m \le M$.

Inductive step: We need to show that $L_{M+n+1} = F_{M+2}L_n + F_{M+1}L_{n-1}$.

$$L_{M+n+1} = L_{M+n} + L_{M+n-1} = F_{M+1}L_n + F_ML_{n-1} + F_{M+1}L_{n-1} + F_ML_{n-2}$$
$$= F_{M+1}(L_n + L_{n-1}) + F_M(L_{n-1} + L_{n-2}) = F_{M+1}L_{n+1} + F_ML_n$$

Ooops…that's not quite it, let's try again:

$$L_{M+n+1} = L_{M+n} + L_{M+n-1} = (F_{M+1}L_n + F_ML_{n-1}) + (F_ML_n + F_{M-1}L_{n-1})$$
$$= (F_{M+1} + F_M)L_n + (F_M + F_{M-1})L_{n-1} = F_{M+2}L_n + F_{M+1}L_{n-1}$$

c. $F_{n-1} + F_{n+1} = L_n$

Proof: Base case: If *n* = 2 then we have $F_1 + F_3 = 1 + 2 = 3 = L_2$, which is true.

Inductive hypothesis: Assume $F_{n-1} + F_{n+1} = L_n$ for $1 \le n \le N$, then

Inductive step: We need to show that $F_N + F_{N+2} = L_{N+1}$.

$$F_N + F_{N+2} = (F_{N-1} + F_{N-2}) + (F_{N+1} + F_N) = (F_{N-2} + F_N) + (F_{N-1} + F_{N+1}) = L_{N-1} + L_N$$
$$= L_{N+1}$$

d. $L_{n-1} + L_{n+1} = 5F_n$

Proof: Base case: If *n* = 2 then $L_1 + L_3 = 1 + 4 = 5F_2 = 5 \cdot 1$ is true.

Inductive hypothesis: Assume that $L_{n-1} + L_{n+1} = 5F_n$ for $2 \le n \le N$.

Inductive step: We want to show that $L_N + L_{N+2} = 5F_{N+1}$

$$L_N + L_{N+2} = (L_{N-1} + L_{N-2}) + (L_{N+1} + L_N) = (L_{N-1} + L_{N+1}) + (L_{N-2} + L_N)$$
$$= 5F_N + 5F_{N-1} = 5(F_N + F_{N-1}) = 5\,F_{N+1}$$

e. $F_nL_n = F_{2n}$

Proof: Base case: If *n* = 1 then $F_nL_n = F_{2n} \Leftrightarrow 1 \cdot 1 = 1$ is true.

Inductive hypothesis: Assume that $F_nL_n = F_{2n}$ for $1 \le n \le N$,

Inductive step: We want to show that $F_{N+1}L_{N+1} = F_{2(N+1)}$.

$F_{N+1}L_{N+1} = (F_{N+1})(F_N + F_{N+2}) =$ (by the result of part (c) above)

$= F_{N+1}F_N + F_{N+1}\,F_{N+2}$

Now recall from part (a) $F_{m+n} = F_mF_{n-1} + F_{m+1}F_n$ and substitute *m* = *N* + 1 and *n* = *N* + 1 to get $F_{2(N+1)}F_{N+1}F_N + F_{N+1}\,F_{N+2}$, as desired.

f. $F_n^2 = F_{n-1}F_{n+1} + (-1)^{n-1}$

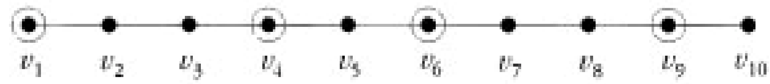
Proof: Base case: $F_2^2 = F_1F_3 + (-1)^1 = 2 - 1 = 1$ is true.

Inductive hypothesis: Assume that $F_n^2 = F_{n-1}F_{n+1} + (-1)^{n-1}$ is true for $1 \le n \le N$.

Inductive step: We would like to show that $F_{N+1}^2 = F_NF_{N+2} + (-1)^N$

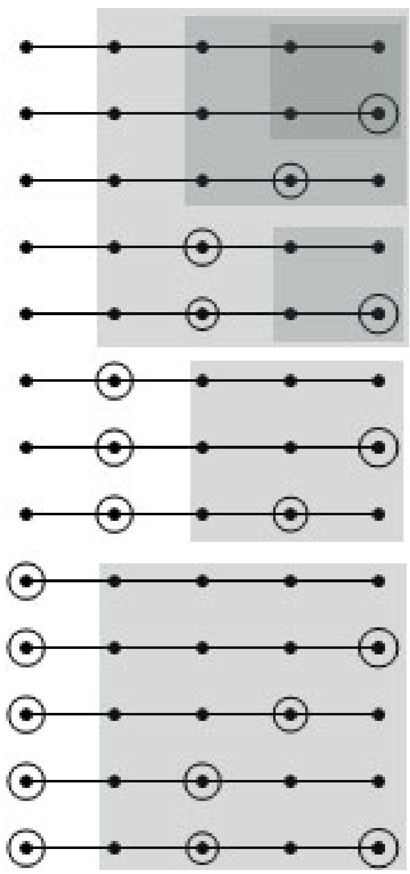
$F_{N+1}^2 - F_NF_{N+2} = F_{N+1}(F_N + F_{N-1}) - F_N(F_{N+1} + F_N) = F_{N+1}F_N + F_{N+1}F_{N-1} - F_NF_{N+1} -$
$F_N^2 = F_{N+1}F_{N-1} - F_N^2 = -(-1)^{N-1}$ and the result follows.

6. The following problems are related.
   a. Let a_n denote the number of length *n* binary sequence with no consecutive 0's. For instance, 0110101101. Show that a_n satisfies the recurrence relation, $a_n = a_{n-1} + a_{n-2}$.
   SOLN: Any suck sequence starting with a 1 can be extended in a_{n-1} ways and any such sequence starting with a 0 must be followed by a 1 and then can be extended in a_{n-2} ways. Thus a_n grows like the Fibonacci numbers.
   b. Let b_n denote the number of ways to select a subset of nonadjacent vertices from a path on *n* vertices (as in the figure below, where nonadjacent vertices v_1, v_4, v_6, v_9 are chosen.) Such a subset of vertices is called an *independent set*. Notice that b_{n-2} and b_{n-1} count the number of independent subsets that do and do not contain the first point on the path, respectively. Show that $b_n = F_{n+2}$, a Fibonacci number.



   This establishes a correspondence between independent sets of vertices and the 0's of binary sequences. Note that 0110101101 from part (a) corresponds to the independent set above.

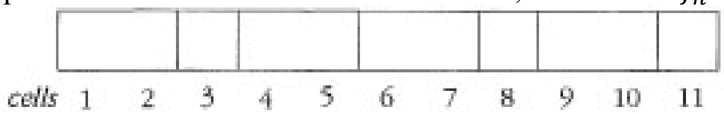
   SOLN: b_1 is the number of ways to select a subset of nonadjacent vertices from a path with one vertex. There are two ways to do this: either select it, or don't select it. In the figure at right, these correspond to the two top vertices in the right most column. So $b_1 = 2 = F_3$.
   For b_2 you can preface each of the two existing sets with an uncircled node and then add a third path which can be thought of as the first uncricled node preceded by a circled node. This leads to three independent sets with two vertices and so $b_2 = b_1 + 1 = 3 = F_4$. Now each of these three paths can be preceded by an uncircled node and the paths of b_2 which don't start with a circled node (that is, the 2 paths of b_1 which were prefaced by an uncircled node) can be prefaced with a circled node, leading to $b_3 = b_2 + b_1 = 3 + 2 = 5 = F_5$. Growth continues in this fashion so that b_{n-1} and b_{n-2} count the number of independent subsets that do and do not contain the first point in the path, respectively. Thus $b_n = b_{n-1} + b_{n-2}$, the Fibonacci recurrence relation, and since we've shown that initial values of the sequence match subsequent Fibonacci values, there is a 1-1 correspondence between the numbers b_n of independent sets and the number of sequences a_n described in part (a).



   c. Note that there is a natural correspondence between these two representations, a_n and b_n: independent sets of vertices correspond to 0's in in the binary sequences. Let c_n denote the number of series of 1's and 2's that add to *n*. Then $c_1 = 1$ and $c_2 = 2$ since 1 = 1 is the only way to represent 1 and 2 = 2 = 1+1 are the two ways to represent 2. Prove that $c_n = F_{n+1}$.
   SOLN: By conditioning on the first number in the sum, we see the Fibonacci recursion formula, $c_n = c_{n-1} + c_{n-2}$. That is, if the first number in the sum is a 1, then the rest of the numbers must add up to *n* – 1, and there are c_{n-1} such sums, and if the first number is a 2, there are c_{n-2} such sums. Since the first numbers are $c_1 = 1$ and $c_2 = 2$, if follows that $c_n = F_{n+1}$.

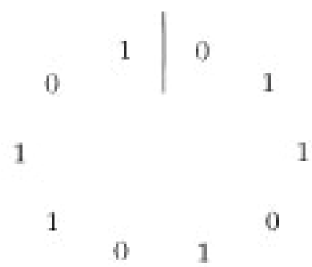
Note that there is a natural correspondence between c_{n+1} and b_n. For a given series of 1's and 2's that add to $n + 1$, associate the subset of vertices whose *indices* are not partial sums of the series. For example, the series 2 + 1 + 2 + 2 + 1 + 2 + 1 = 11 has partial sums 2, 3, 5, 7, 8, 10 and 11 yielding the independent set v_1, v_4, v_6, v_9.

d. Now consider the number of ways to tile a $1 \times n$ checkerboard with cells labeled $1, 2, \dots, n$. Let f_n denote the number of ways to tile an n-board with 1×1 *squares* and 1×2 *dominoes*. Associating each square with a 1 and each domino with a 2, we see that $f_n = c_n$.

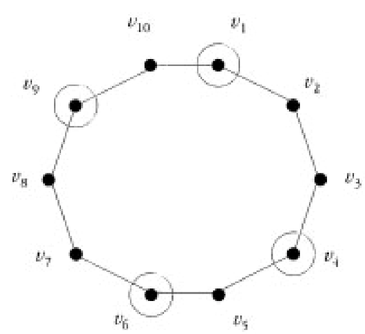


Thus, $f_n = F_{n+1}$. So f_{m+n} is the number of ways to tile a length $m + n$ board. Explain why $f_{m+n} = f_m f_n + f_{m-1} f_{n-1}$.
SOLN: The left side, f_{m+n}, counts the number of ways to tile a length $m + n$ board. To interpret the right side, note that tilings come in two varieties: either they can separated into a length *m* tiling followed by a length *n* tiling, or they cannot. There are $f_m f_n$ tilings of the first type. Tilings of the second type must contain a domino covering cells *m* and *m*+1. Ther remaining board can be covered in $f_{m-1} f_{n-1}$ ways.

7. Lucas numbers act like Fibonacci numbers running in circles.
   a. Let A_n denote the number of length *n circular* binary sequences with no consecutive 0's (as in the figure.) What are the length 2 and 3 circular sequences?
   SOLN: The length 2 and 3 circular sequences are 01, 10, 11, 011, 101, 110, and 111.



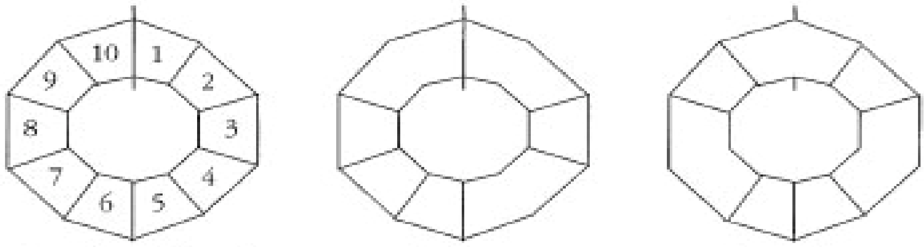
   b. Prove that $A_n = L_n$, the *n*th Lucas number.
   SOLN: We've established that $A_2 = 3 = L_2$ and $A_3 = 4 = L_3$. To prove that $A_n = L_n$, we show that it satisfies the Fibonacci recurrence relation. Condition on the first digit: A circular sequence beginning with 1 can be completed in $a_{n-1} = F_{n+1}$ ways (see #6a). A 0 must be surrounded by 1's, so that a circular sequence starting with 0 can be completed in $a_{n-3} = F_{n-1}$ ways. Thus $A_n = F_{n+1} + F_{n-1} = L_n$ (see #5c).



   c. Let $B_n = A_n = L_n$ denote the number of independent sets in a cycle graph with *n* vertices.
   Let C_n denote the number of series of 1's and 2's that sum to *n* with the end point restriction that it may not begin and end with a 2. Show that $C_n = L_{n-1}$.

   SOLN: By conditioning on the first term, we have (using the sequence $\{c_n\}$ from #6c and #5c) that $C_n = c_{n-1} + c_{n-3} = F_n + F_{n-2} = L_{n-1}$. That is, if the first term is a 1, then there are c_{n-1} ways to choose the remaining 1's and 2's (no worries about ending with a 2) but if the first term is a 2, then we must end with a 1, so there are c_{n-3} ways to choose the remaining 1's and 2's.

d. Let l_n denote the number of ways to tile a circular $1 \times n$ board with squares and dominoes. Cells are labeled 1 through *n* and a tiling is called an *n*-bracelet. (see below.)
An *n*-bracelet is *out of phase* if a domino covers cells *n* and 1, otherwise the *n*-bracelet is *in phase*.



Circular 10-board   In phase   Out of phase

Show that the number of in phase *n*-bracelets is $f_n = F_{n+1}$ and the number of out of phase *n*-bracelets is $f_{n-2} = F_{n-1}$ and that $l_n = L_n$.
SOLN: For the in-phase case we can tile as if it were a linear board, so there are F_{n+1} ways (#6d). For the out of phase case we have two spaces fixed, so there are 2 less squares that have options for filling, meaning there are F_{n-1} tilings in that case. Thus there are $L_n = F_{n-1} + F_{n+1}$ tilings.

8. Consider the following function *p*, where *L* is a list.
B. If *L* = *x*, a single element, then *p*(*L*) = "*x*".
R. If *L* = *L'*, *x* for some list *L'*, then *p*(*L*) = "*x*, *p*(*L'*)".
If *L* = `john, paul, george, ringo`, what is *p*(*L*)?
SOLN: Ringo comes first for a change:
$p(L) =$ **`ringo,`** *p*(**`john,paul,george`**)
= **`ringo,george,`** *p*(**`john,paul`**)
= **`ringo,george,paul,`** *p*(**`john`**)
= **`ringo,george,paul,john`**

9. Suppose *L* is an SList of depth *p*. Find a recurrence relation for *A*(*p*), the number of times two numbers are added when evaluating Sum(*L*).
SOLN: Recall the definition of SList:
B. *x* where $x \in \mathbb{R}$
R. (X, Y) where *X* and *Y* are SLists having the same number of elements and the last element of *X* is less than the first element of *Y*.

Also, define of Sum(*L*):
B. If *L* = *x*, a single number, then Sum(*L*) = *x*
R. If $L = (X, Y)$ where *X* and *Y* are SLists then Sum(*L*) = Sum(*X*) + Sum(*Y*).

Claim: The number of times two numbers are added together when evaluating Sum(*L*) is

$$A(p) = \begin{cases} 0 & \text{if } p = 0 \\ 2 \cdot A(p-1) + 1 & \text{if p} > 0 \end{cases}$$

Proof: If the depth of the list is 0, then there is just one number and no additions are involved. If the depth of the list is 1, then $L = \{x_1, x_2\}$ so $A(1) = 2 \cdot A(0) + 1 = 1$ addition is needed. Now suppose *X* and *Y* are lists of depth *N* , each requiring $A(N)$ additions, then Sum(*X*,*Y*) = Sum(*X*) + Sum(*Y*) requires $2 \cdot A(N)$ sums to add the *X* and *Y* lists, plus 1 more to add those sums together. QED . Is there a closed form for this? Yeah: $A(p) = \sum_{n=0}^{p-1} 2^n = 2^p - 1$.

10. Let L be an SList. Define a recursive function Wham as follows.

B. Suppose $L = x$. Then $\text{Wham}(L) = x \cdot x$.

R. Suppose $L = (X, Y)$. Then
$\text{Wham}(L) = \text{Wham}(X) + \text{Wham}(Y)$.

a. Evaluate $\text{Wham}\big((2,4)(6,7)\big)$. Remember to show all work.

SOLN: $\text{Wham}\big((2,4)(6,7)\big) = \text{Wham}(2,4) + \text{Wham}(6,7)$
$= \text{Wham}(2) + \text{Wham}(4) + \text{Wham}(6) + \text{Wham}(7) = 2^2 + 4^2 + 6^2 + 7^2 = 105$

b. Give a recurrence relation for $S(p)$, the number of + operations performed by Wham on an SList of depth p, for $p \geq 0$.

SOLN: This is essentially the same question as #9, since the only thing that's changed is that instead of $\text{Sum}(x) = x$, we have $\text{Wham}(x) = x^2$. So the answer is the same.

c. Give a recurrence relation for $M(p)$, the number of · operations performed by Wham on an SList of depth p, for $p \geq 0$.

$$M(p) = f(x) = \begin{cases} 1, & p = 0 \\ 2 \cdot M(p-1), & p > 0 \end{cases}$$

Which has closed form $M(p) = 2^p$.

11. An urn contains six red balls, six white balls, and six blue balls, and sample of three balls is drawn at random without replacement. Compute the probability that the sample contains at least (that is, exactly) one ball of each color. (Round your answer to four decimal places.)

SOLN: There are $\binom{18}{3} = \frac{18 \cdot 17 \cdot 16}{3!} = 816$ equally likely outcomes from choosing 3 balls from 18 without replacement and without regard to order. Only $\binom{3}{3} + \binom{3}{2} \cdot 2 + \binom{3}{1} \cdot 3 = 10$ of these are distinguishable outcomes, however:
(R,W,B),(R,R,W),(R,W,W),(R,R,B),(R,B,B),(W,W,B)(W,B,B),(R,R,R),(W,W,W),and(B,B,B).
How many ways are there to choose exactly one red, one white and one blue? There are 6 ways to choose a red, 6 ways to choose a blue and 6 ways to choose a white, so there are $6 \cdot 6 \cdot 6 = 216$ ways to choose a exactly one of each color. So the probability we seek is $\frac{6 \cdot 6 \cdot 6}{\frac{18 \cdot 17 \cdot 16}{3!}} = \frac{6 \cdot 6 \cdot 6 \cdot 6}{18 \cdot 17 \cdot 16} = \frac{9}{34} \approx 0.2647 = 26.47\%$.

To check this you can write a simulator like this:

```
#include <iostream>
#include <ctime>
#include <cstdlib>
const int samples = 1000000;
using namespace std;
int main() {
   int first, second, third, success = 0;
   int urn[18] = {0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2};
   srand(time(0));
   for(int i = 0; i < rolls; i++) {
         first = rand()%18;
         do second = rand()%18; while(first==second);
         do third = rand()%18; while(third==first || third == second);
         if(urn[first] != urn[second] &&
            urn[first] != urn[third] &&
            urn[second] != urn[third]) ++success;
   }
   cout << (float)success/samples;     return 0;
}
```

Whose output verifies our conjecture: 0.264454

12. An urn contains two red balls and five blue balls. Draw two balls at random from the urn, without replacement. Compute the expected number of red balls in your sample. (Round your answer to four decimal places.)

SOLN: There are $\binom{7}{2} = 21$ equally likely outcomes when drawing without replacement.

Expected number of red balls = $\sum_{n=0}^{2} n \cdot \Pr(n) = \Pr(1 \text{ red ball}) + 2 \cdot \Pr(2 \text{ red balls})$

There are 2 ways to choose a red ball and for each of those, 5 ways to choose a blue ball, so there are $5 \cdot 2 = 10$ different ways to choose 1 red and 1 blue ball. Thus, $\Pr(1 \text{ red ball}) = \frac{2 \cdot 5}{\binom{7}{2}} = \frac{10}{21}$ and $\Pr(2 \text{ red balls}) = \frac{1}{21}$ so the expected number of red balls is $\frac{12}{21} = \frac{4}{7} \approx 0.5714$

13. Consider the following algorithm.

```
for i ∈ {1, 2, 3, 4} do
  ┌ beep
    For j ∈ {1, 2, 3} do
      beep
    for k ∈ {1, 2, 3, 4} do
      ┌ for l ∈ {1, 2, 3, 4, 5, 6} do
          beep
        for m ∈ {1, 2, 3, 4, 5} do
  └   └   beep
```

How many times does a **beep** statement get executed?

SOLN: $4 + 4 \cdot 3 + 4 \cdot 4 \cdot (6 + 5) = 4 \cdot (1 + 3 + 44) = 4 \cdot 48 = 192$

14. Let $x_1, x_2, \ldots, x_n$ be an array. Consider the following algorithm.

```
for i ∈ {1, 2, ... , ⌊n/2⌋} do
  ┌ t ← x_i
    x_i ← x_{n-i+1}
  └ x_{n-i+1} ← t
```

a. How many $\leftarrow$ operations does this algorithm perform? Your answer should be a function of *n*.

SOLN: $3 \cdot \left\lfloor \frac{n}{2} \right\rfloor$

b. What does this algorithm do to the array?

SOLN: It continually swaps the ith element with the n+1-ith element starting with the first and continuing up until, but not including the middle, effectively reversing the array.

15. An urn contains *m* red balls and *n* green balls.

a. Give a big-Θ estimate for the number of ways to draw a sequence of *n* green balls without replacement.

SOLN: This is the number of permutations on *n*: $\Theta(n!)$.

b. Give a big-Θ estimate for the number of ways to choose 2 red balls and 3 green balls (assuming $m > 1$ and $n > 2$) without replacement.

SOLN: This is polynomial in *n* and *m*: $\Theta(n^2 m^3)$