1. (•) ***Write the sentence "I understand the course policies."***

   Solution: I understand the course policies. ■

   > **Rubric:** ***No credit for HW0 without this sentence!*** (If you forgot, please write the sentence on the first page of your HW1 solutions.)

   (a) Solve these recurrences.

   - $A(n) = 4A(n-1) + 1$
   - $B(n) = B(n-3) + n^2$
   - $C(n) = 2\,C(n/2) + 3\,C(n/3) + n^2$
   - $D(n) = 2\,D(n/3) + \sqrt{n}$
   - $E(n) = \begin{cases} n & \text{if } n \le 3, \\ \dfrac{E(n-1)E(n-2)}{E(n-3)} & \text{otherwise} \end{cases}$

   Solution: Here is everything you needed to write for full credit:

   $$\boxed{\begin{array}{l} A(n) = \Theta(4^n) \\ B(n) = \Theta(n^3) \\ C(n) = \Theta(n^2) \\ D(n) = \Theta(n^{\log_3 2}) \\ E(n) = \Theta(3^{n/2}) \end{array}}$$

   And here are the proofs; see the recurrences handout for more details.

   - $A(n) = \Theta(4^n)$ — The recurrence unrolls into the geometric series $\sum_{i=1}^{n} 4^i$. Only the largest term in a geometric series matters for $\Theta(\cdot)$ bounds.
   - $B(n) = \Theta(n^3)$ — The recurrence unrolls into the sum $B(n) = \sum_{i=\Theta(1)}^{\Theta(n)} \Theta(i^2)$.
   - $C(n) = \Theta(n^2)$ — Using recursion trees; the level sums define a decreasing geometric series.
   - $D(n) = \Theta(n^{\log_3 2})$ — Using recursion trees; the level sums define an increasing geometric series.
   - $E(n) = \Theta(3^{n/2})$ — The first several values of this function are $1, 2, 3, 6, 9, 18, 27, 54, 81, 108, 243, 324, \ldots$; every other term is a power of 3. The function satisfies the simpler recurrence $E(n) = 3 \cdot E(n-2)$, which unrolls into a geometric series.

   ■

   > **Rubric:** 5 points max: 1 point each. −½ for approximations like $D(n) = \Theta(n^{0.63093})$ or $E(n) = \Theta(1.73205^n)$ instead of exact answers. No proofs are necessary.

(b) Sort these functions from asymptotically smallest to asymptotically largest, indicating ties if any.

n	$\lg n$	$\sqrt{n}$	7^n
$\sqrt{\lg n}$	$\lg \sqrt{n}$	$7^{\sqrt{n}}$	$\sqrt{7^n}$
$7^{\lg n}$	$\lg(7^n)$	$7^{\lg \sqrt{n}}$	$7^{\sqrt{\lg n}}$
$\sqrt{7^{\lg n}}$	$\lg(7^{\sqrt{n}})$	$\lg \sqrt{7^n}$	$\sqrt{\lg(7^n)}$

Solution:

$$
\begin{aligned}
\sqrt{\lg n} \ll \lg \sqrt{n} \equiv \lg n \\
&\ll 7^{\sqrt{\lg n}} \\
&\ll \sqrt{n} \equiv \sqrt{\lg(7^n)} \equiv \lg(7^{\sqrt{n}}) \\
&\ll n \equiv \lg(7^n) \equiv \lg \sqrt{7^n} \equiv 7^{\log_7 n} \\
&\ll 7^{\lg \sqrt{n}} = \sqrt{7^{\lg n}} = n^{(\lg 7)/2} \\
&\ll 7^{\lg n} = n^{\lg 7} \\
&\ll 7^{\sqrt{n}} \\
&\ll \sqrt{7^n} = (\sqrt{7})^n \\
&\ll 7^n
\end{aligned}
$$

■

Rubric: 5 points max. −½ for each misplaced, missing, or repeated function, but no negative scores. No proofs are necessary.

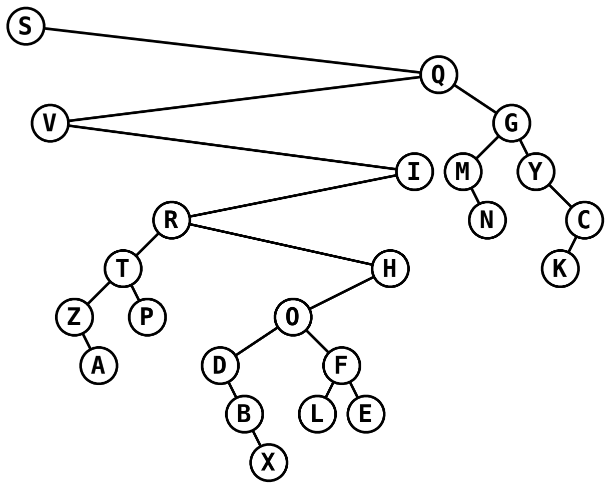
2. (a) List the nodes in Prof. della Giungla's tree in the order visited by a *preorder* traversal.

Solution: **S Q V I R T Z A P H O D B X F L E G M N Y C K** ■

Rubric: 5 points max. −1 for each misplaced, missing, extra, or repeated letter, but no negative scores.

(b) Draw Prof. della Giungla's tree.

Solution:



■

Rubric: 5 points max. −1 for each misplaced, missing, extra, or repeated node, but no negative scores.

3. Describe a data structure that stores a set of n points in the plane and supports the queries HIGHESTTORIGHT and RIGHTMOSTABOVE.

Solution: Let S be the given set of points. Say that a point (a, b) is *dominated* by another point (x, y) if $a < x$ and $b < y$. Call a point in S *extreme* if it is not dominated by any other point in S. We can make two interesting observations about extreme points.

First, HIGHESTTORIGHT and RIGHTMOSTABOVE always return extreme points or NONE. Thus, there is no reason to store non-extreme points.

Second, suppose (a, b) and (c, d) are extreme points. If $a < c$, then $b > d$. Thus, sorting the extreme points by increasing x-coordinate is the same as sorting them by decreasing y-coordinate.

The data structure is a simple array containing the extreme points of S, sorted in order by increasing x-coordinate (or equivalently, by decreasing y-coordinate). This data structure clearly requires $O(n)$ space.

We can answer HIGHESTTORIGHT(ℓ) in $O(\log n)$ time as follows. If the last x-coordinate in the array is smaller than ℓ, return NONE. Otherwise, find the first point in the array whose x-coordinate is greater than ℓ by binary search.

We can answer RIGHTMOSTABOVE(ℓ) in $O(\log n)$ time as follows. If the first y-coordinate in the array is smaller than ℓ, return NONE. Otherwise, find the last point in the array whose y-coordinate is greater than ℓ by binary search. ■

Rubric: 10 points max = 4 for data structure + 1 for space analysis + 2 for HIGHESTTORIGHT algorithm + 2 for RIGHTMOSTABOVE algorithm + 1 for time analysis. Full credit for any $O(n)$-space, $O(\log n)$-time solution. +3 extra credit for storing each point at most once.

Partial credit for correct but slower/larger solutions. Max 4 points total for any solution with $\Omega(n)$ query time (for example, using a binary search tree without specifying that it is balanced); scale partial credit.

4. Prove that for any arithmetic expression tree, there is an equivalent expression tree in normal form.

Solution (structural induction): Before we prove anything, let's establish some notation. Suppose A and B are expression trees. Let $(A+B)$ denote the expression tree whose root is a $+$-node, whose left subtree is A, and whose right subtree is B. Similarly, let $(A \times B)$ denote the expression tree whose root is a $\times$-node, whose left subtree is A, and whose right subtree is B. For any arithmetic expression tree T, let f_T denote the function it represents. The definitions imply immediately that $f_{(A+B)} = f_A + f_B$ and $f_{(A\times B)} = f_A \times f_B$.

We first prove an auxiliary result and then a useful special case.

Lemma 1. *For any arithmetic expression trees L and R in normal form, the expression tree $(L+R)$ is in normal form.*

Proof: Let v be an arbitrary $+$-node in $(L+R)$ that is not the root. If the parent of v is in L, then it must be a $+$-node, because L is in normal form. Similarly, if the parent of v is in R, then it must be a $+$-node, because R is in normal form. The only other possibility is that the parent of v is the root of $(L+R)$, which is a $+$-node. We conclude that every $+$-node in $(L+R)$ either has no parent or is the child of another $+$-node. Thus, $(L+R)$ is in normal form. □

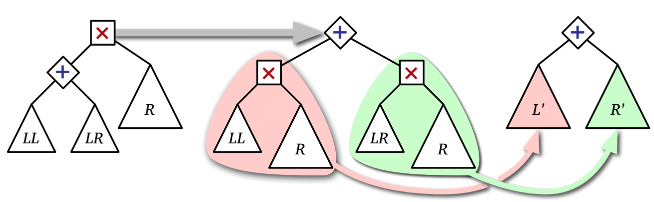
Lemma 2. *For any arithmetic expression trees L and R in normal form, there is an arithmetic expression tree in normal form that is equivalent to $(L \times R)$.*

Proof: Let L and R be arbitrary arithmetic expression trees in normal form. Assume that for every proper subtree L' of L, there is an arithmetic expression tree in normal form that is equivalent to $(L' \times R)$. There are three cases to consider:

- Suppose subtrees L and R have no $+$-nodes. (For example, L might be a single variable node.) Then the expression tree $(L \times R)$ has no $+$-nodes, and is therefore vacuously in normal form.
- Suppose subtree L contains a $+$-node. Then the root of L must be a $+$-node, because otherwise L would not be in normal form. Let LL and LR be the left and right subtrees of L, respectively, so $L = (LL + LR)$. Define a new expression tree $\boldsymbol{T := ((LL \times R) + (LR \times R))}$. We easily verify that T is equivalent to $(L \times R)$, as follows:

$$\begin{aligned} f_T \;=\; f_{((LL\times R)+(LR\times R))} \;&=\; f_{(LL\times R)} + f_{(LR\times R)} \;=\; (f_{LL}\times f_R) + (f_{LR}\times f_R) \\ &=\; (f_{LL}+f_{LR})\times f_R \;=\; f_{(LL+RR)}\times f_R \;=\; f_{((LL+LR)\times R} \;=\; f_{(L\times R)}. \end{aligned}$$

  Because LL is a proper subtree of L, the inductive hypothesis implies that there is a normal-form expression tree L' equivalent to $(LL \times R)$. Similarly, because LR is a proper subtree of L, the inductive hypothesis implies that there is a normal-form expression tree R' equivalent to $(LR \times R)$. Define a new expression tree $\boldsymbol{T' := (L' + R')}$. Because L' and R' are in normal form, Lemma 1 implies that T' is in normal form. We can easily verify as above that T' is equivalent to T, and therefore equivalent to $(L \times R)$.



- Finally, suppose R contains a $+$-node, but A does not. The expression tree $(R \times L)$ is clearly equivalent to $(L \times R)$. The previous case implies that there is an arithmetic expression tree in normal form that is equivalent to $(R \times L)$, and therefore to $(L \times R)$.

In all cases, we conclude that there is an arithmetic expression tree in normal form that is equivalent to $(L \times R)$. □

Theorem. *For any arithmetic expression tree T, there is an arithmetic expression tree in normal form that is equivalent to T.*

Proof: Let T be an arbitrary arithmetic expression tree. Assume that for any proper subtree S of T, there is an arithmetic expression tree in normal form that is equivalent to S. There are three cases to consider: the root of T is labeled with a variable, $+$, or $\times$.

- If T is a single variable node, then T is already in normal form.
- Suppose $T = (L + R)$ for some expression trees L and R. Because L and R are proper subtrees of T, the inductive hypothesis implies that there is a normal-form expression tree L' that is equivalent to L, and there is an normal-form expression tree R' that is equivalent to R. Define a new expression tree $\boldsymbol{T' := (L' + R')}$.
  The inductive hypothesis implies that $f_{L'} = f_L$ and $f_{R'} = f_R$. Thus,
  $$f_{T'} \;=\; f_{(L'+R')} \;=\; f_{L'} + f_{R'} \;=\; f_L + f_R \;=\; f_{(L+R)} \;=\; f_T;$$
  in other words, T' is equivalent to T. Because both L' and R' are in normal form, Lemma 1 implies that $T' = (L' + R')$ is also in normal form.
- Finally, suppose $T = (L \times R)$ for some expression trees L and R. Because L and R are proper subtrees of T, the inductive hypothesis implies that there is a normal-form expression tree L' that is equivalent to L, and there is an normal-form expression tree R' that is equivalent to R. It is easy to check that $(L' \times R')$ is equivalent to T, exactly as in the previous case. Lemma 2 implies that there is an expression tree T' in normal form that is equivalent to $(L' \times R')$, and therefore equivalent to T.

In all cases, we conclude that there is an arithmetic expression tree in normal form that is equivalent to T. □

■

Rubric: 10 points max = 1 for base case(s) + 3 for the $(L + R)$ case + 6 for the $(L \times R)$ case. This proof is *much* more verbose than required for full credit. This is not the only correct proof; however, most correct arguments do require two separate induction proofs.

Max 4 points for a 'proof' of the $(L \times R)$ case that tries to recurse on a *larger* expression tree, without an explicit proof that the recursion terminates. (For example: "Apply the distributive law repeatedly; eventually the tree is in normal form." There *is* a well-founded partial order over the set of all expression trees that makes this argument work, but its description is quite technical.) Solutions that make this mistake can still get partial credit for the base case and the $(L + R)$ case.

Zero points for any 'induction' argument of the form "Suppose some expression tree T can be converted to normal form; now add something to T get a new tree T'...", without an ***explicit, self-contained*** argument that the proof considers ***all*** arithmetic expression trees, ***even if the base case is correct***. This is not a mistake; it's a sin.

5. (a) What is the *exact* expected number of cards that Professor Jay hurls into the watermelon?
   (b) Give *exact* probabilities for four statements about the *first* pair of cards.
   (c) Give *exact* probabilities for four statements about the *last* pair of cards.

Solution: Here is everything you needed to write for full credit:

(a) 1751/52	(b) i. 1/169 ii. 103/2704 or 6/169 iii. 11/16 iv. 78/169	(c) i. 7/103 ii. 31/103 or 24/103 iii. 77/103 iv. 48/103

Parts (b)ii and (c)ii are ambiguous and therefore have two correct answers. Here are the proofs:

(a) Suppose each deck has n cards; I'll plug in $n = 52$ at the end. Let R denote the number of cards in the shuffled red deck that are *above* $\boxed{3♣}$; for example, $R = 0$ if $\boxed{3♣}$ is the top card in the red deck, and $R = n - 1$ if $\boxed{3♣}$ is the bottom card in the red deck. Similarly, let B denote the number of cards above $\boxed{3♣}$ in the shuffled blue deck. The random variables R and B are independent and uniformly distributed over the range $\{0, 1, 2, \ldots, n-1\}$.

Now let $W = \min\{R, B\}$; this is the number of *pairs* of cards Prof. Jay hurls into the watermelon. We need to compute $\mathrm{E}[2W] = 2 \cdot \mathrm{E}[W]$.

$$\begin{aligned}
\mathrm{E}[W] &= \sum_{i=0}^{n-1} i \cdot \Pr[W = i] = \sum_{i=1}^{n-1} \Pr[W \ge i] && \text{[definition of E[ ]]} \\
&= \sum_{i=1}^{n-1} \Pr[R \ge i \text{ and } B \ge i] && \text{[definition of } W\text{]} \\
&= \sum_{i=1}^{n-1} \Pr[R \ge i] \cdot \Pr[B \ge i] && \text{[}R \text{ and } B \text{ are independent]} \\
&= \sum_{i=1}^{n-1} \frac{n-i}{n} \cdot \frac{n-i}{n} && \text{[}R \text{ and } B \text{ are uniform]} \\
&= \sum_{j=1}^{n-1} \frac{j^2}{n^2} && \text{[substitute } j = n - i\text{]} \\
&= \frac{(n-1)n(2n-1)}{6n^2} && \text{[closed form for } \textstyle\sum j^2\text{]} \\
&= \frac{(n-1)(2n-1)}{6n} && \text{[algebra]}
\end{aligned}$$

(The sum-of-squares identity can be found on Wikipedia at http://en.wikipedia.org/wiki/Square_pyramidal_number.) We conclude that the expected number of cards hurled into the watermelon is exactly $2 \cdot \mathrm{E}[W] = (n-1)(2n-1)/3n$, which is just less than one-third of the $2n$ cards. Finally, plugging in $n = 52$ gives us the answer $(51 \cdot 103)/(3 \cdot 52) = (17 \cdot 103)/52 = 1751/52$.

(b) There are exactly $52^2 = 2704$ different possibilities for the first pair of cards, and all these possibilities are equally likely. Thus, the probability that *any* statement about the first pair of cards is true is an integer multiple of $1/2704$.

i. *Both cards are threes:* There are $13^2 = 169$ possible pairs of ranks, and only one of them is a pair of 3s.

ii. *One card is a three, and the other card is a club:* This statement is ambiguous!

- If we read this statement as "***At least*** one card is a three...", it can be satisfied by 103 different pairs: $4 \cdot 13$ where the red card is a $\boxed{3}$ and the blue card is a $\boxed{\clubsuit}$, plus $13 \cdot 4$ where the red card is a $\boxed{\clubsuit}$ and the blue card is a $\boxed{3}$, minus 1 for the pair $\boxed{3\clubsuit}$ $\boxed{3\clubsuit}$, which we've counted twice.

- If we read it as "***Exactly*** one card is a three...", it can be satisfied by 96 different pairs: $4 \cdot 12$ where the red card is a $\boxed{3}$ and the blue card is a $\boxed{\clubsuit}$ *but not the* $\boxed{3\clubsuit}$, plus $12 \cdot 4$ where the blue card is a $\boxed{\clubsuit}$ and the red card is a $\boxed{3}$ *but not the* $\boxed{3\clubsuit}$.

iii. *If (at least) one card is a heart, then (at least) one card is a diamond:* We can ignore the ranks and focus on the $4^2 = 16$ choices for the suits. The statement is logically equivalent to "Neither suit is ♥ or at least one suit is ♦." There are exactly five suit pairs that make this statement ***false***:

$\boxed{\heartsuit}\,\boxed{\spadesuit} \qquad \boxed{\heartsuit}\,\boxed{\clubsuit} \qquad \boxed{\heartsuit}\,\boxed{\heartsuit} \qquad \boxed{\clubsuit}\,\boxed{\heartsuit} \qquad \boxed{\spadesuit}\,\boxed{\heartsuit}$

Thus, the statement is true with probability $11/16$.

iv. *The card from the red deck has higher rank than the card from the blue deck:* We can ignore the suits, and focus on the $13^2 = 169$ choices for the ranks. There are 13 pairs of equal ranks; in exactly half of the remaining 156 pairs, the red card has higher rank. So there are 78 pairs where the red rank is higher than the blue rank.

(c) There are exactly $52 + 52 - 1 = 103$ different possibilities for the last pair of cards, all equally likely: 52 where the red card is the $\boxed{3\clubsuit}$, plus 52 where the blue card is a $\boxed{3\clubsuit}$, minus 1 for the pair $\boxed{3\clubsuit}$ $\boxed{3\clubsuit}$. Thus, the probability that *any* statement about the last pair of cards is true is an integer multiple of $1/103$.

i. *Both cards are threes:* There are exactly seven possibilities:

$\boxed{3\spadesuit}\,\boxed{3\clubsuit} \quad \boxed{3\heartsuit}\,\boxed{3\clubsuit} \quad \boxed{3\diamondsuit}\,\boxed{3\clubsuit} \quad \boxed{3\clubsuit}\,\boxed{3\clubsuit} \quad \boxed{3\clubsuit}\,\boxed{3\diamondsuit} \quad \boxed{3\clubsuit}\,\boxed{3\heartsuit} \quad \boxed{3\clubsuit}\,\boxed{3\spadesuit}$

Thus, the probability that both cards are threes is $7/103$.

ii. *One card is a three, and the other card is a club:* Again, the statement has two interpretations:

- ***At least*** *one...:* If the red card is $\boxed{3\clubsuit}$, there are exactly 16 blue cards that make the statement true: Any $\boxed{3}$ or any $\boxed{\clubsuit}$. Similarly, if the blue card is $\boxed{3\clubsuit}$, there are exactly 16 red cards that make the statement true. But we've counted the pair $\boxed{3\clubsuit}$ $\boxed{3\clubsuit}$ twice. Thus, the probability is exactly $31/103$.

- ***Exactly*** *one...:* If the red card is $\boxed{3\clubsuit}$, there are exactly 12 blue cards that make the statement true: any $\boxed{\clubsuit}$ except the $\boxed{3\clubsuit}$. Similarly, if the blue card is $\boxed{3\clubsuit}$, there are exactly 12 red cards that make the statement true. Thus, the probability is exactly 24/103.

iii. *If (at least) one card is a heart, then (at least) one card is a diamond:* There are exactly 26 possible pairs that make the statement ***false***: $\boxed{?\heartsuit}$ $\boxed{3\clubsuit}$ and $\boxed{3\clubsuit}$ $\boxed{?\heartsuit}$. Thus, the statement is true with probability 77/103.

iv. *The card from the red deck has higher rank than the card from the blue deck.* There are exactly seven pairs where both cards have the same rank; see part i above. The red card has higher rank in exactly half of the remaining 96 pairs. Thus, the red card has higher rank with probability 48/103.

■