# menu-chapter-errorsCommon Student Mistakes

**Author:** Kenneth H. Rosen, AT&T Laboratories.

## 0102153menu-chapter-errors25520451Introduction

Here are some common mistakes that people studying discrete mathematics sometimes make. The list is organized chapter by chapter, based on when they first occur, but sometimes mistakes made early in the course perpetuate in later chapters. Also, some of these mistakes are remnants of misconceptions from high school mathematics (such as the impulse to assume that every operation distributes over every other operation).

In most cases we describe the mistake, give a concrete example, and then offer advice about how to avoid it. Note that additional advice about common mistakes in given, implicitly or explicitly, in the solutions to the odd-numbered exercises, which constitute the bulk of the *http://www.mhhe.com/math/advmath/rosen/Student Solutions Guide for Discrete Mathematics and Its Applications*, fourth edition, by Kenneth H. Rosen.

## 0102153menu-chapter-errors25520451Solving Problems in Discrete Mathematics

Before getting to the common mistakes, we offer some general advice about solving problems in mathematics, which are particularly relevant to working in discrete mathematics. The problem-solving process should consist of the following steps. (This four-step approach is usually attributed to the mathematician George Pólya (1887–1985).)

1. Read and understand the problem at hand. Play around with it to get a feeling for what is going on and what is being asked.

2. Apply one or more problem-solving strategies to attack the problem. Do not give up when one particular tactic doesn't work. This phase of the process can take a long time.

3. Carefully write up the solution when you have solved the problem. Make sure to communicate your ideas clearly.

4. Look back on what you have done. Make sure that your answer is correct (think of creative ways to test it). Also consider other ways of solving the problem, and think about how you might generalize your results.

Your bag of problem-solving strategies may include drawing a picture or diagram; looking at special cases or simpler instances of your problem; looking at related problems; searching for patterns; making tables of what you know (and in general, being organized); giving names to what you don't know and writing equations about it (or, more generally, applying the mathematical tools you have learned in algebra and other courses); working backwards and setting subgoals ("I could solve this problem if I could do such-and-such"); using trial and error (usually called "guess and check" in educational circles); using indirect reasoning and looking for counterexamples ("if this weren't true, then what would have to happen?"); jumping out of the system and trying something totally different; or just going away from the problem and coming back to it later. You will find it useful to read over this description of the problem-solving process repeatedly during your study of discrete mathematics.

Here is an example of applying the problem-solving process to a problem in discrete mathematics. Suppose we want to find the number of squares (of all sizes) on a chessboard. We understand this to mean not only the obvious 64 little squares, but also the $2 \times 2$ squares (of which there will be many), the $3 \times 3$ squares, and so on, up to the entire $8 \times 8$ board itself. We should draw a picture to see what's going on here. Let's begin by playing with a smaller version of this problem, using a board of size $2 \times 2$. Then obviously there are 4 small squares and the entire board, for a total of $4 + 1 = 5$. Let's try a $3 \times 3$ board. Here there are 9 little squares, and it is easy to see that there are 4 squares of size $2 \times 2$ (nuzzled in the upper left, the upper right, the lower left, or the lower right), as well as the entire board; so the answer here is $9 + 4 + 1 = 14$. A pattern seems to be emerging—we seem to be adding perfect squares to get the answer. Maybe for the $4 \times 4$ board there will be $16+9+4+1 = 30$ squares in all. We draw the $4 \times 4$ picture and verify that this is correct. In fact, we can see exactly what is going on, the way the upper left corner of a $k \times k$ square can be in any of the first $5 - k$ rows and first $5 - k$ columns of the board, for $k = 1, 2, 3, 4$, and so there are $(5 - k)^2$ squares of size $k$ in the $4 \times 4$ board, exactly as our sum indicated. We have now solved the problem and can write up a solution explaining why the answer is $8^2+7^2+6^2+5^2+4^2+3^2+2^2+1^2 = 204$. In the looking back stage (step 4) we would certainly want to notice that for an $n \times n$ board, there are $\sum_{i=1}^{n} i^2$ squares. To continue our investigation, we might want to explore such further questions as allowing rectangles rather than squares, looking at rectangular chessboards rather than square ones (or counting triangles in boards made up of triangles), or moving to 3-dimensional space and counting cubes in a large cube. Notice how we followed the process outlined above and used many of the strategies listed.

---

## **0102153menu-chapter-errors25520451**List of Common Mistakes

If students or instructors have items to add to the lists below, please let the author know.

*0102153menu-chapter-errorsChapter 1*

- *Incorrectly translating English statements involving implication into symbolic form.* There are many errors of this type. For example, there are difficulties with the use of the word "or" in English; be sure to differentiate between inclusive and exclusive versions (see page 5 of the text). An implication is quite different from a conjunction, but some speakers fail to distinguish them; to say that $B$ will happen *if* $A$ happens is quite different from saying that $A$ and/or $B$ *will* happen. Perhaps the most common mistake is confusing $p \to q$ with $q \to p$. To say, for example, that I will go to the movie *if* I finish my homework means something quite different from asserting that I will go to the movie *only if* I finish my homework.

- *Incorrectly negating compound statements without using De Morgan's laws—in effect saying, for example, that $\neg(p \vee q)$ is logically equivalent to $\neg p \vee \neg q$, or that $\neg(p \wedge q)$ is logically equivalent to $\neg p \wedge \neg q$.* For example, if it is not true that John is over 18 years old or lives away from home, then it is true that he is not over 18 years old *and* (not *or*) he does not live away from home. The correct statements are that $\neg(p \vee q)$ is logically equivalent to $\neg p \wedge \neg q$, and that $\neg(p \wedge q)$ is logically equivalent to $\neg p \vee \neg q$. This mistake is a general instance of assuming that every operation distributes over every other operation, here that negation distributes over disjunction (or conjunction).

- *Incorrectly forming complements of sets without using De Morgan's laws—in effect saying, for example, that $\overline{A \cap B} = \overline{A} \cap \overline{B}$.* The correct statements are $\overline{A \cap B} = \overline{A} \cup \overline{B}$ and $\overline{A \cup B} = \overline{A} \cap \overline{B}$. This is another general instance of assuming that every operation distributes over every other operation, here that complementation distributes over intersection (or union). Students sometimes make similar errors in algebra, such as asserting, incorrectly, that $\sqrt{a^2 + b^2} = a + b$ or $\sin(\alpha + \beta) = \sin\alpha + \sin\beta$.

- *Misinterpreting the meaning of the word "any" in a mathematical statement.* This word is ambiguous in many situations, and so should usually be avoided in mathematical writing. If you are not sure whether the writer meant "every" or "some" when the word "any" was used, get the statement clarified. As a corollary, of course, you should avoid using this word yourself. Here is an example: What would one mean if she defined a purple set of integers to be one "in which any integer in the set has at least three distinct prime divisors"? Does "any" mean "every" here (in which case the set $\{30, 40\}$ is not purple), or does "any" mean "some" here (in which case the set $\{30, 40\}$ is purple)?

- *Incorrectly writing the symbolic form of an existential statement as $\exists x(A(x) \rightarrow B(x))$ instead of $\exists x(A(x) \wedge B(x))$.* For example, the symbolic form of "There exists an even number that is prime" is $\exists x(E(x) \wedge P(x))$, not $\exists x(E(x) \rightarrow P(x))$, where we are letting $E(x)$ mean "$x$ is even" and $P(x)$ mean "$x$ is prime." As a rule of thumb, existential quantifiers are usually followed by conjunctions.

- *Incorrectly writing the symbolic form of a universal statement as $\forall x(A(x) \wedge B(x))$ instead of $\forall x(A(x) \rightarrow B(x))$.* For example, the symbolic form of "Every odd number is prime" is $\forall x(O(x) \rightarrow P(x))$, not $\forall x(O(x) \wedge P(x))$, where we are letting $O(x)$ mean "$x$ is odd" and $P(x)$ mean "$x$ is prime." As a rule of thumb, universal quantifiers are usually followed by implications.

- *Incorrectly putting predicates inside predicates, such as $P(O(x))$.* For example, if $P(x)$ means "$x$ is prime," and $O(x)$ means "$x$ is odd," then it would never make sense to write $P(O(x))$ in trying to express a statement such as "$x$ is an odd prime" or to write $\forall x\, P(O(x))$ to say "all odd numbers are prime." The notation $P(O(x))$ would mean that *the assertion* that $x$ is odd is a prime number, and clearly an assertion isn't any kind of number at all. Functional notation has a wonderful internal beauty and consistency to it—the thing inside the parentheses has to be what the thing outside the parentheses applies to.

- *Failure to change the quantifier when negating a quantified proposition, especially in English.* For example, the negation of the statement that some cats like liver is *not* the statement that some cats do not like liver; it is that no cats like liver, or that all cats dislike liver.

- *Using incorrect notation regarding elements and subsets of power sets and confusing the notions of "element" and "subset" when dealing with a power set.* If $A$ is a subset of $S$, then $A$ is an element of the power set of $S$. For example, if $S = \{p, q, r\}$, then $\{p, r\} \subseteq S$, so $\{p, r\} \in P(S)$. On the other hand, $\{p, r\} \nsubseteq P(S)$, and $\{p, r\} \notin S$. Also, note that $\emptyset \notin S$, $\emptyset \in P(S)$, and $\{\emptyset\} \notin P(S)$.

- *Incorrectly writing* $\{\emptyset\}$ *to represent the empty set.* One reason this cannot be the empty set is that it has one element in it. Correct notation for the empty set is either $\{\ \}$ or $\emptyset$. It is a set with no elements in it.

- *Incorrectly omitting parentheses in expressions involving intersection, union, and difference of sets.* In absence of a default order of operations, an expression such as $A \cap B \cup C$ is ambiguous, since it might mean either $(A \cap B) \cup C$ or $A \cap (B \cup C)$, and these are not the same sets. It is important to put in parentheses so that the reader knows what you mean.

- *Erroneously believing that all infinite sets are countable.* Although it is possible to give an infinite list of certain infinite sets (such as the set of rational numbers, using an order like $0, \frac{1}{1}, -\frac{1}{1}, \frac{1}{2}, -\frac{1}{2}, \frac{2}{1}, -\frac{2}{1}, \frac{1}{3}, -\frac{1}{3}, \frac{3}{1}, -\frac{3}{1}, \frac{1}{4}, -\frac{1}{4}, \frac{2}{3}, -\frac{2}{3}, \frac{3}{2}, -\frac{3}{2}, \frac{4}{1}, -\frac{4}{1}, \frac{1}{5}, \ldots$, it is not possible to do this for certain larger sets, such as the set of real numbers. There is no list of all the real numbers, as was proved in Section 1.7.

- *Incorrectly applying to infinite sets intuition that is valid for finite sets.* For example, if a *finite set* $A$ can be put into one-to-one correspondence with a proper subset of $B$, then clearly $|A| < |B|$. This is not true if $A$ is infinite, however: Let $A$ be the even natural numbers and let $B$ be the natural numbers—then $A$ can be put into one-to-one correspondence with (in fact *is*) a proper subset of $B$, but $A$ can also be put into one-to-one correspondence with all of $B$ (pairing $2n$ with $n$ for $n = 0, 1, 2, \ldots$), so in fact $|A| = |B|$.

- *Failing to regard a function as an object at a higher order of abstraction than an element, an ordered pair, or a set.* The notation $f : A \to B$ means that $f$ is a process—a rule that must apply to every element of $A$ and in each case yield a result in $B$. The function is not $A$, it is not $B$, it is not an element of $A$ or $B$, nor is it the action on just one element of $A$. For example, if $f$ is the function from $\{1, 2, 3\}$ to the natural numbers that has the rule $f(x) = x^2$, then $f$ is the entire process that takes 1 to 1, 2 to 4, and 3 to 9; it is not, for example, just the number 9 or just the pair $(3, 9)$.

- *Confusing the idea that a function must be well defined with the concept of one-to-oneness.* A function in general need not have the property that distinct elements of the domain be sent to distinct elements of the range. What a function must satisfy is the requirement that two different elements of the range cannot both be the image of the same element of the domain. Thus $f(x) = x^2$ *is* a function from the set of real numbers to the set of nonnegative real numbers (it's onto but not one-to-one), but $f(x) = \pm\sqrt{x}$ is not a function from the nonnegative real numbers to the real numbers.

- *Incorrectly calculating values of floor and ceiling functions, especially for negative values.* The floor function always rounds down, and the ceiling function always rounds up. Thus, for example, $\lfloor -3.2 \rfloor = -4$, and $\lceil -3.2 \rceil = -3$.

- *Incorrectly including diagonal portions in sketches of floor and ceiling functions and their variations.* These graphs almost always consist entirely of horizontal segments.

- *Thinking of a big-O estimate as if it were a big-Theta estimate, i.e., thinking that is provides both a lower bound and an upper bound on the size of the function.* Big-$O$ estimates are only upper bounds. It is not correct to say that such-and-such an algorithm is inefficient because it runs in time $O(n^7)$, since, for instance, a linear algorithm (one with running time proportional to $n$) also satisfies this estimate. What one would want to say in such a case (if it is true) is that the algorithm runs in time $\Theta(n^7)$.

- *Not understanding that the constant in a big-O estimate can be very large.* For example, if an algorithm has running time $O(n^2)$, then it might take $10^{55}n^2$ steps and so be impractical even for small values of $n$.

- *Being misled by exponents when comparing dissimilar functions.* For example, it might at first glance look as if $(\log n)^{100}$ is growing faster than $n^{1/2}$, but in fact the latter is growing faster, since $\log n$ grows so slowly compared to $n$. Thus $(\log n)^{100}$ is $O(n^{1/2})$, but $n^{1/2}$ is not $O((\log n)^{100})$.

*0102153menu-chapter-errorsChapter 2*

- *Incorrectly making am negative.* For example, $-165$ is 4, not $-1$.

- *Forgetting to increment counters inside loops of procedures when constructing algorithms.* If the counter is not incremented, the loop will probably run forever.

- *Adding when one should be multiplying, or vice versa, when analyzing running times of algorithms.* If one loop is nested inside another, then the running times are multiplied; if one loop follows another, the times are added.

- *Confusing $a/b$ and $a|b$.* The slanted slash is an operation, and the result of the operation is a *number*. For example, $6/3$ is the number 2. The vertical bar is the *verb* of a sentence. For example, $3|6$ is asserting that 3 is a divisor of 6; it is not speaking of the result of actually carrying out the division.

- *Writing $a|b$ when one means $b|a$.* It is true that $3|6$, but it is not true that $6|3$.
- *Incorrectly considering 1 to be a prime number.* It's as much a matter of convention as anything else, but the number 1 is, by definition, neither prime nor composite.
- *Forgetting that all positive integers are divisors of 0, and therefore that $\gcd(a, 0) = a$ for all positive integers $a$.* For example, $\gcd(6, 0) = 6$. Of course 0 is not a divisor of any nonzero number, and division by 0 is undefined.
- *When carrying out the Euclidean algorithm, incorrectly using the last quotient as the final answer (the* gcd*), rather than the last divisor.* For example, if the last step is to divide 4 into 8, giving a quotient of 2 and remainder of 0, then the gcd is the final divisor, 4, not the final quotient, 2.
- *Incorrectly assuming that everything that works for equality works for congruence.* For example, it is not true that just because $r$ and $s$ are congruent modulo $m$, $a^r \equiv a^s \pmod{m}$. Try $m = 3$, $a = 2$, $r = 1$, and $s = 4$. Also, note that even though $8 \equiv 14 \pmod{6}$, it would be wrong to divide both sides by 2 and claim that $4 \equiv 7 \pmod{6}$.
- *Incorrectly assuming that multiplication of matrices means entry-by-entry multiplication.* Addition of matrices is done term by term, but multiplication is more complex.

*0102153menu-chapter-errorsChapter 3*

- *Overusing the term "by definition" in justifying statements in a proof.* For example, Franklin Roosevelt was not the President of the United States at the start of the country's entry into World War II in December, 1941, "by definition"; he was the President because he had been elected to that position in 1940 and had not died or left office.
- *Not going back to carefully check the definitions in justifying statements in a proof.* For example, if one is trying to prove something about the divisibility relation ($a|b$) on integers, then it is important to correctly use the meaning of that relation (that $b = ac$ for some integer $c$) at one or more places in the proof.
- *Incorrectly starting a proof by assuming what is to be proved.* A common occurrence of this is an earlier course is trying to prove trigonometric identities by starting with the identity and using algebra to reach $A = A$; this is not valid. Similarly, if we are trying to prove a set identity, such as $A \subseteq A \cup B$, it would be invalid to start with the statement $A \subseteq A \cup B$.

- *Invalidly assuming that a few (or even a large number of) examples of a universally quantified proposition imply that the proposition is true.* There is an example from number theory of an intriguing proposition about a positive integer $n$ that is true for every $n \leq 1{,}000{,}000$ with the sole exception of $n = 1969$. A proof of the universally quantified proposition $\forall x\, P(x)$ consists of showing that the property $P(x)$ holds no matter what $x$ is chosen from the universe of discourse.

- *Incorrectly assuming that an arbitrary element has a particular property when all you know is that there exists an element with the property.* For example, suppose we are trying to prove the assertion that $x^2$ always leaves a remainder of 1 when divided by 8. It would be invalid to start our proof by assuming that $x = 2n + 1$ for some integer $n$; even though *some* integers have this property of being odd, it is not true that all of them do, so we would be proving the assertion to be true only some of the time, rather than always.

- *Forgetting to do the basis case in a proof by mathematical induction.* The inductive step goes through fine, for example, if we try to prove that $n = n + 1$ for all positive integers $n$, but this proposition is obviously not true. The catch is that the basis case (when $n = 1$) fails, since $1 \neq 1 + 1$.

- *Failing to do more than one basis case in a proof by mathematical induction in certain situations, such as when the inductive step needs two or more previous conditions.* For example, when proving statements about the Fibonacci sequence, it usually is necessary to check the first two basis cases (say $n = 1$ and $n = 2$), since the inductive step relies on the equation $f_n = f_{n-1} + f_{n-2}$.

- *Confusing a summation with the propositional function $P(n)$ in an induction proof.* For example, in trying to prove $1+2+3+\cdots+n = n(n+1)/2$ by induction, $P(n)$ is this entire equation, not its left-hand side.

- *Not being organized when attempting to write a recursive definition.* Good advice here is to think about how you want to build up the items under discussion, step by step. The inductive rules of the definition need to be formulated to permit each such step, and base cases are needed to get the process off the ground. Common mistakes include not including enough base cases (for example, the recursive definition of regular expressions in Chapter 10 requires three base cases), having conflicting cases (for example, having one clause to handle $n$ divisible by 2 and another clause to handle $n$ divisible by 3, and thereby not having a unique definition for those $n$ divisible by 6), or having a function value at $n$ depend on a function value at an input larger than $n$ (e.g., trying to set $f(n) = f(3n+1) - 2$).

- *In a proof by mathematical induction, writing $P(n+1)$ incorrectly.* Once $P(n)$ is properly formulated, writing $P(n+1)$ can usually be done more or less mechanically by plugging $n+1$ in for $n$. For example, if $P(n)$ is the statement that $2+4+6+\cdots+n = n(n+1)$, then $P(n+1)$ is the statement that $2+4+6+\cdots+n+(n+1) = (n+1)(n+2)$.

- *In a proof by mathematical induction, making errors in basic algebra, especially in simplifying expressions.* For example, you might have to use the fact that $2^n + 2^n = 2^{n+1}$, or to simplify $(n+1)^3 + 5(n+1)^2$, which is best done by factoring, not by first expanding each term. Carefully check your algebraic manipulations when you have trouble with the inductive step of such a proof.

- *In a recursive algorithm, failing to let the computer do the recursing.* The hardest thing to overcome in thinking about recursive algorithms is the reluctance to believe that the smaller case will be handled correctly. For example, suppose that you want to write a recursive algorithm to compute $a^n$, using the facts that $a^{2k} = (a^k)^2$ and $a^{2k+1} = (a^k)^2 \cdot a$. The recursive call will handle the calculation of $a^k$, and so you don't need to worry about how the computer will repeatedly recurse, all the way down to the base case, in order to do that. As long as your recursive step and basis case (here, $a^0 = 1$) are correct, your algorithm is correct.

*0102153menu-chapter-errorsChapter 4*

- *Drawing an incorrect diagram when solving counting problems.* Diagrams are very useful in all of mathematics, and drawing a diagram is almost always a good way to start solving a problem; thus *failing to draw a diagram* could also be considered a common mistake. For example, you should draw a row of six blanks (not five) as a template for constructing words of length 6 whose symbols are chosen from a set of five elements. Tree diagrams are also sometimes quite helpful.

- *Not determining whether or not order matters in solving a counting problem.* For example, if we are asked for the number of ways to write 7 as the sum of positive integers, then we need to know whether $3+2+2$ and $2+3+2$ are to be considered the same way or distinct ways. Read the problem very carefully to understand what is being counted. Resolve any ambiguities ahead of time by explicitly stating any assumptions that seem to be missing from the problem formulation.

- *Not determining whether or not repetitions are allowed in solving a counting problem.* For example, if we are asked for the number of ways to choose five donuts from a shop selling eight varieties of donuts, we need to know whether we are allowed to choose more than one donut of the same variety. Read the problem very carefully to understand what is being counted. Resolve any ambiguities ahead of time by explicitly stating any assumptions that seem to be missing from the problem formulation.

- *Counting each item in the set under discussion more than once, not recognizing that an adjustment needs to be made for double counting.* For example, if we count handshakes person by person, then we need to recognize that each shake has been counted twice, once for each of its participants.

- *Counting each item in the set under discussion more than once, not recognizing that the inclusion–exclusion principle is needed.* For example, if we are told that there are 26 computer science majors and 34 mathematics majors at a certain university, then there may not be $26 + 34 = 60$ people majoring in either computer science or mathematics, since these two numbers might both include the double majors. To correctly compute the total number of people majoring in these subjects, we would need to subtract the number of double majors from this sum.

- *Using the pigeonhole or generalized pigeonhole principle incorrectly.* It helps to explicitly identify the pigeons and the holes. For example, to find the minimum number of cards that must be chosen in order to guarantee that at least six of the same suit are picked, the holes are the suits, and the cards are the pigeons (the answer is 21).

- *When trying to calculate the probability that one of two events will occur, incorrectly taking the sum of the individual probabilities.* For example, the probability that a 3 will show up if a fair die is rolled twice is not $\frac{1}{6} + \frac{1}{6}$, the sum of the probability that the 3 occurs on the first roll and the probability that the 3 occurs on the second roll. Instead, we should calculate this as 1 minus the probability that the 3 fails to appear on either roll (which is $\frac{5}{6} \cdot \frac{5}{6}$, since the rolls are independent). Thus the correct answer is $1 - \frac{25}{36} = \frac{11}{36}$, rather than $\frac{2}{6}$.

- *Assuming that all events in a probability calculation are disjoint.* Doing so can lead to absurd conclusions, such as a probability greater than 1. (This is really a generalization of the previously listed mistake.) For example, to calculate the probability that someone else in your graduating class of 400 students shares your birthday (assuming that all birthdays are equally likely and ignoring February 29), you cannot argue that since each of them has a probability of 1/365 of sharing your birthday, the probability is 399/365 (probabilities can never exceed 1, and in any case, this event is not a certainty).

- *Assuming that all events in a probability calculation are independent.* For example, to calculate the probability that we get two hearts when drawing two cards from a deck of cards, without replacing the first card before drawing the second, we cannot simply note that the probability of drawing a heart is $\frac{13}{52}$ on each draw (that much is true) and therefore conclude that the answer is $\frac{13}{52} \cdot \frac{13}{52} = \frac{1}{16}$. Instead, we must determine that for the second draw, the probability of drawing a heart, given that we drew a heart on the first draw, is $\frac{12}{51}$, and therefore that the probability of drawing a heart both times is $\frac{13}{52} \cdot \frac{12}{51} = \frac{1}{17}$.

- *Getting misled by the subtle assumptions inherent in probability problems.* The most famous example here is the Monty Hall Three Door Problem (see page 265 of the text). Unless one is very careful about the assumptions one makes about the game host's protocol, one cannot calculate the probability that switching doors will change your chances of winning. For example, if the host (who knows where the prize lies) were to offer you a switch if and only if you had chosen the correct door, then obviously it would be wrong for you to switch when he makes the offer. A national debate about this problem raged for many months when it was popularized in a magazine article.

- *Letting intuition interfere with reason in working with probability.* For example, it might seem counter-intuitive that among a group of 23 people, the odds favor two of them having the same birthday, but the calculation shows this to be true.

*0102153menu-chapter-errorsChapter 5*

- *Failing to note the need for the inclusion–exclusion principle.* To believe that $|A \cup B| = |A| + |B|$ is always true is related to the wishful thinking that every operation distributes (or otherwise behaves in some simple, agreeable way) with respect to every other operation. This equality holds only when $A$ and $B$ are disjoint.

- *Confusing the signs of the terms when applying the inclusion–exclusion principle.* Note that the signs alternate as we take larger and larger unions.

- *Not including all the terms when applying the inclusion–exclusion principle.* If there are $n$ sets involved, then there are nearly $2^n$ different terms in the equation altogether.

- *Giving up too easily when trying to write down a recurrence relation to model a problem situation.* Ask yourself how one can obtain an instance of the problem of size $n$ from instances of sizes $n-1$ (or sometimes also smaller instances). Make sure to consider all the possibilities, and make sure to include enough initial conditions. For example, if $a_n$ is the number of ways to climb $n$ stairs if we are allowed to take them either one at a time or three at a time, then clearly $a_1 = 1$, $a_2 = 1$, and $a_3 = 2$, and then $a_n = a_{n-1} + a_{n-3}$ for $n \geq 4$, since the first step could be a single step or a triple step.

- *Misapplying the algorithm for solving linear homogeneous recurrence relations with constant coefficients when there are repeated roots of the characteristic equation.* One needs to multiply by powers of $n$ in this case. For example, if the characteristic equation is $r^2 - 6r + 9 = (r-3)^2 = 0$, then the general solution is $a_n = c_1 \cdot 3^n + c_2 n \cdot 3^n$.

- *Finding a bogus particular solution of a linear nonhomogeneous recurrence relation with constant coefficients.* It is always advisable to check the solutions you obtain. For example, if you had computed that $a_n = 2^n$ was a particular solution to $a_n = 2a_{n-1} + 2^n$, then plugging this in would show you that you must have made an error, since it is not true that $2^n = 2 \cdot 2^{n-1} + 2^n$.

- *Forgetting to use the inclusion–exclusion principle when counting solutions to an equation in nonnegative integers.* For example, to count the number of solutions to $x + y + z = 58$ where $0 \leq x < 8$, $0 \leq y < 10$, and $0 \leq z < 15$, one needs to count the number of solutions when the upper bound restrictions are lifted, then subtract the number of solutions in which each such restriction is violated, then add back the number of solutions in which two such restrictions are violated simultaneously, and finally subtract the number of solutions in which all three restrictions are violated.

- *Forgetting to worry about the first few terms of a power series.* When solving a recurrence relation by using generating functions, the recurrence relation usually kicks in only for $k \geq 1$ or 2; thus the first term or two must be handled explicitly.

- *Failing to change the variable in a power series when necessary.* For example, if a power series has $x^{k-1}$ and you need it to be $x^k$, you can replace $k$ by $k+1$ throughout the summation (including the limits) and simplify algebraically: $\sum_{k=1}^{\infty} k\, x^{k-1} = \sum_{k+1=1}^{\infty} (k+1)\, x^{(k+1)-1} = \sum_{k=0}^{\infty} (k+1)\, x^k$.

- *Setting up the wrong model when solving counting problems with generating functions.* You need to carefully work out what each factor of the generating function needs to be, worrying about how much repetition is allowed and whether order matters. See, for instance, Example 12 in Section 5.4 of the text, where the proper generating function depends on whether or not we take order into account.

- *Making algebraic errors in working with generating functions.* When expanding a generating function to find the coefficient of $x^n$, one must of course use the distributive law. The algebraic manipulations can get messy, as the number of terms can grow rapidly. One solution to this problem is to use a computer algebra package such as maple to do the algebra. For example, to multiply out $(1+x+x^2)(1+x^2+x^4+x^6)$, you end up with 12 terms, which then simplify to $1+x+2x^2+x^3+2x^4+x^5+2x^6+x^7+x^8$.

- *Not knowing how to use partial fraction decomposition when dealing with generating functions, or making errors in the procedure, such as forgetting to include terms of the form $(x-a)^k$ for all $k$ such that $1 \leq k \leq n$ when the factor $(x-a)^n$ appears in the denominator of the fraction to be expanded.* This subject is traditionally taught in calculus courses, even though it has little to do with calculus (other than the fact that it is used as a technique of integration). Therefore those students who have not yet studied enough calculus (partial fractions are usually covered in the second semester), or who have taken a course in which this topic is not covered, may need to find a source of instruction for this useful tool (or rely on a computer algebra package such as maple to perform the task). Any traditional calculus text will probably have a section from which this material can be learned or reviewed.

*0102153menu-chapter-errorsChapter 6*

- *Failing to draw a picture when dealing with relations.* The digraph of a relation on a set gives an excellent way to visualize what is going on. This common mistake can be generalized: Failing to draw a picture when dealing with any mathematical object. See the list of general problem-solving strategies given in the introduction to this section of the *Guide*.

- *Forgetting to think about pairs $(a,b)$ and $(b,a)$ when checking for transitivity of a relation or forming the transitive closure.* In this case, one needs to have (or add) the loops $(a,a)$ and $(b,b)$ as well.

- *Failing to recognize that symmetry or transitivity often hold vacuously.* For example, the relation $\{(1,2),(1,3)\}$ is transitive, because it is vacuously true that whenever $(a,b)$ and $(b,c)$ are in the relation, so is $(a,c)$—i.e., there are no pairs making the hypothesis of this implication true.

- *Forgetting that order of operations matters when forming closures.* The symmetric, transitive closure is not the same as the transitive, symmetric closure, for example.

- *Incorrectly assuming that every relation has desired properties, such as symmetry or transitivity.* It is certainly not true that every relation is a total order or an equivalence relation. Many partial orders are not total, so elements can be incomparable (for example, it is true neither that $\{1,2\} \subseteq \{1,3\}$ nor that $\{1,3\} \subseteq \{1,2\}$). If I know you and you know Mary, that doesn't mean that I necessarily know Mary. If $f(x)$ is $O(g(x))$, that doesn't mean that $g(x)$ is $O(f(x))$.

- *Invalidly applying to infinite partially ordered sets intuition that is valid for finite posets.* Hasse diagrams can be misleading for infinite posets (or they may not exist at all), so be wary about thinking of all posets in finite diagrammatic terms. For example, in the poset consisting of the positive real numbers under the $\leq$ relation, there is no immediate successor to any element, so there would be no edges in the Hasse diagram if we were to try to draw one.

- *Forgetting to eliminate all the implied edges when drawing a Hasse diagram.* If there is an edge from $a$ to $b$ and one from $b$ to $c$, where $a$ lies above $b$ and $b$ lies above $c$, then you must not show an edge between $a$ and $c$.

- *Forgetting that certain pairs in a relation are implied in a Hasse diagram.* If there is an edge from $a$ to $b$ and one from $b$ to $c$, where $a$ lies above $b$ and $b$ lies above $c$, then $c$ is related to $a$ even though there is no edge between $a$ and $c$.

- *Incorrectly supposing that all partial orders have least or greatest elements, or that least upper bounds or greatest lower bounds always exist.* For example, in the poset $(\{a,b,c,d,e\},\{(a,a),(b,b),(c,c),(d,d),(e,e),(a,b),(c,b),(c,d),(e,d)\})$ there is no least element, no greatest element, and no least upper bound for $\{a,e\}$. Similarly, in the poset consisting of the integers under $\leq$ together with two extra elements $x$ and $y$ that are defined to be less than all the integers and unrelated to each other, there is no least upper bound for $\{x,y\}$, even though every integer is an upper bound.

*0102153menu-chapter-errorsChapter 7*

- *Getting confused by some of the terminology in graph theory, such as the distinction between path and simple path.* Here is one place where memorization is required. Making your own glossary on file cards or in a computer file may be helpful.

- *Overcounting the edges in a graph by forgetting to divide by 2 when adding the degrees of the vertices.* Each edge is counted twice, once for each end.

- *Being unsure about what kind of graph model to use.* If the relationship between objects in the situation you are trying to model is symmetric, then an undirected graph is probably appropriate; otherwise a directed graph usually works best. For example, highways joining major cities can be traversed in either direction, so an undirected graph is appropriate for this model. The predator-prey relation among species of animals is definitely not symmetric, so a digraph seems right here.

- *Incorrectly thinking that if two graphs share many of the same attributes (invariants like number of vertices, number of edges, etc.) then they must be isomorphic.* See Section 7.3 for some counterexamples.

- *Incorrectly interchanging the definitions of Hamilton and Euler paths and circuits.* Remember that there is a simple test for Euler paths and circuits, but no one knows of any simple tests for Hamilton paths and circuits.

- *Ignoring the fact that having an Euler [Hamilton] circuit implies the existence of an Euler [Hamilton] path.* Look carefully at the definitions.

- *Confusing theorems in graph theory with their converses.* For example, if in a connected simple graph with $n \geq 3$ vertices each vertex has degree at least $n/2$, then the graph has a Hamilton circuit; but the converse or inverse of this statement is not true (there are plenty of graphs having Hamilton circuits in which the vertex degrees are small). Here is another example: If a connected simple graph is planar, then it must satisfy $e \leq 3v - 6$, where $e$ is the number of edges and $v$ is the number of vertices. Therefore (by the contrapositive), we know that if a graph has too many edges ($e > 3v - 6$), then it cannot be planar. What we *cannot* conclude is the converse—it is not a theorem that if $e \leq 3v - 6$, then the graph has to be planar.

- *Using the nonexistent word "vertice" instead of the correct word "vertex" to talk about just one of the dots in a graph.* Similarly, there is no such thing as a "matrice."

- *Mistakenly thinking that a graph is nonplanar just because it is drawn one way with two edges crossing.* If it is possible to redraw the graph without edges crossing, then the graph is planar. For example, $K_4$ is planar, even though drawing it as the vertices of a square with straight line segments representing the edges causes a crossing in the middle of the picture. (Redraw it as the vertices of a triangle with one more vertex in the interior.)

- *Invalidly concluding that once one has found a coloring of a graph with $n$ colors, its chromatic number has to be $n$.* In fact, all we know in that case is that its chromatic number is at most $n$. It may be possible to find another coloring with fewer than $n$ colors. For example, one could color $C_4$ with four colors (a different color for each vertex), but its chromatic number is in fact 2.

- *Mistakenly believing that greedy algorithms always produce the optimal solution to a problem.* It often happens that the simple-minded greedy approach *does* find the best solution (e.g., in looking for minimum spanning trees in Section 8.6), but often the greedy approach fails (e.g., in finding a coloring of a graph using as few colors as possible).

- *Failing to recognize that writing down a procedure doesn't guarantee that it does what you want it to do.* For example, one cannot write down a greedy algorithm to color a graph and then claim without justification that this procedure finds the coloring with the fewest possible colors. In fact, it won't, as fairly simple counterexamples can show.

*0102153menu-chapter-errorsChapter 8*

- *Incorrectly setting up a decision tree for a problem such as identifying counterfeit coins by weighing them, and thereby drawing the wrong conclusions.* Each possible situation must correspond to a path from the root of the tree to a leaf.

- *Not realizing what type of tree is needed for a particular mathematical model.* Issues to consider are whether there should be a root (a starting point for some process), whether the children of a vertex are ordered, and whether each child should be classified as a right child or a left child.

- *Incorrectly omitting parentheses in expressions written in infix notation.* In absence of a default order of operations, an expression such as $A \cap B \cup C$ is ambiguous, since it might mean either $(A \cap B) \cup C$ or $A \cap (B \cup C)$, and these are not the same sets. With prefix or postfix notation, no such ambiguities arise.

- *Forgetting that when doing an inorder traversal of an ordered rooted tree that is not binary, the root of each subtree comes after the first subtree but before all the other subtrees.* For a binary tree, inorder traversal is rather obvious—left, root, right. When the tree isn't binary, we can still define inorder traversals, but the definition isn't as natural.

- *When applying Prim's algorithm for finding minimum spanning trees, forgetting that edges become eligible for inclusion in the tree gradually (as opposed to Kruskal's algorithm, in which they are all eligible from the start).* If there is a low-cost edge that does not currently have any endpoint in the tree constructed so far, then it cannot yet be added to the tree. When one of its endpoints finally becomes part of the tree, it suddenly becomes eligible and can then be added to the tree if it is the lowest cost edge currently eligible. It is easy to overlook such edges when performing the algorithm.

*0102153menu-chapter-errorsChapter 9*

- *Being off by one level of abstraction when thinking about Boolean functions.* A Boolean function with $n$ variables can be represented by a table with $2^n$ rows; therefore there are $2^{2^n}$ different Boolean functions with $n$ variables.

- *Putting inverters in the wrong place when building combinational circuits.* If we want to invert the value of the output of a gate, the inverter needs to go after the gate.

- *Forgetting to apply De Morgan's laws correctly when evaluating the output of a combinational circuit.* The output of a circuit is a certain Boolean expression of the input variables. When simplifying this expression, it is important to remember than $\overline{x\,y} = \overline{x} + \overline{y}$ and $\overline{x+y} = \overline{x}\,\overline{y}$.

- *Not finding the largest possible blocks when looking for minimum Boolean expressions using Karnaugh maps.* Since there is no known efficient algorithm for solving this problem in general (with more than just a few variables), it should not be surprising that this procedure seems to involve some ugly "guessing" to it.

- *Not finding the best cover when looking for minimum Boolean expressions using the Quine–McCluskey method.* Since there is no known efficient algorithm for solving this problem in general (with more than just a few variables), it should not be surprising that this procedure seems to involve some ugly "guessing" to it. It might be very hard to make sure that a covering we have found with, say, five minterms is really the best possible—that there isn't another covering with four minterms.

*0102153menu-chapter-errorsChapter 10*

- *Incorrectly constructing grammars to generate a desired language.* There is no algorithm for doing this (this statement is a theorem in the theory of computation, similar to Turing's theorem on the unsolvability of the halting problem). Constructing grammars is like writing computer programs, and all the advice given in a programming course (such as thinking from the top down in a structured way) applies.

- *Incorrectly constructing finite-state machines (including Turing machines) to perform a desired task.* There is no algorithm for doing this (this statement is a theorem in the theory of computation, similar to Turing's theorem on the unsolvability of the halting problem). Constructing machines is like writing computer programs, and all the advice given in a programming course (such as thinking from the top down in a structured way) applies.

- *Not including all the strings that are accepted by a given finite-state automaton.* Sometimes students will follow some paths that the machine can take to reach an accepting state and forget to consider others. This will lead to a claim that the language recognized by this automaton is a proper subset of what it really is. Make sure to "play computer" and follow all the branches.

- *Forgetting to have one arrow leaving each state for each input symbol when constructing deterministic finite-state automata.* You usually want to have a "graveyard" state to which the machine goes when it is clear that the input is not acceptable. There needs to be a loop from the graveyard state to itself for each alphabet symbol.

- *Failing to realize that a nondeterministic finite-state automaton can accept a string even when some computation paths on a certain input drive the machine to a nonaccepting state.* As long as at least one path leads to an accepting state, the input string is accepted.

- *Failing to keep track of all the possible states in which a nondeterministic finite-state automaton can enter at each step, when constructing the corresponding deterministic automaton.* Make good use of all your fingers in analyzing what can happen!

- *Failing to check that a machine or a grammar or a regular expression presented as the solution of some problem actually works.* This is similar to debugging a computer program. Many test cases should be tried, so that you can be confident that your machine or grammar or expression really works.

- *When constructing Turing machines, forgetting to include all the cases.* If the machine can ever reach a certain state and be viewing a particular input symbol, then a transition is needed to handle that case. Using top-down programming methodology is advisable to make sure your machines do what you want them to do.

- *Getting so bogged down in the details of constructing Turing machines that you lose sight of the main points of the theory.* The main point is given in the Church–Turing thesis: that every conceivable computation can be performed by any reasonable computational model, be it a Turing machine, your favorite high-level programming language, or yourself working with pencil and paper. And on that note of keeping the "big picture" in mind, we'll bring this list of common mistakes to a close.

menu-chapter-errors chapter:ERRCommon Student Mistakes

IntroductionIntroduction Solving Problems in Discrete MathematicsSolving Problems in Discrete Mathematics

List of Common MistakesList of Common Mistakes Chapter 1Chapter 1 Chapter 2Chapter 2 Chapter 3Chapter 3 Chapter 4Chapter 4 Chapter 5Chapter 5 Chapter 6Chapter 6 Chapter 7Chapter 7 Chapter 8Chapter 8 Chapter 9Chapter 9 Chapter 10Chapter 10