The combinatorial coefficient ${}^nC_k := n!/(k!\,(n-k)!) = {}^nC_{n-k}$. It can be generated in Pascal's triangle by setting ${}^0C_0 = {}^1C_0 = {}^1C_1 = 1$ and then obtaining ${}^{n+1}C_k = {}^nC_k + {}^nC_{k-1}$ recursively:

	k = 0	1	2	3	4	5	6	7	…
n=0:	1								
1:	1	1							
2:	1	2	1						
3:	1	3	3	1					
4:	1	4	6	4	1				
5:	1	5	10	10	5	1			
6:	1	6	15	20	15	6	1		
7:	1	7	21	35	35	21	7	1	

$${}^nC_{k-1} + {}^nC_k = {}^{n+1}C_k$$

Pascal's Triangle

The following three problems are HARD; they are not for everybody.

1 : Prove that if $n > k > 0$ and $GCD(n, k) = 1$, then ${}^nC_k$ is divisible by n . (H.W. Lenstra)

1st Proof: Let $m := n-k > 0$. Since $1 = GCD(n, k) = GCD(m+k, k)$, so is $GCD(m, k) = 1$; this means that integers i and j (not both positive) must exist satisfying $j\cdot m + i\cdot k = 1$. Then

$$\begin{aligned} {}^nC_k = 1\cdot{}^{m+k}C_k &= (j\cdot m + i\cdot k)\cdot{}^{m+k}C_k \\ &= j\cdot m\cdot{}^{m+k}C_m + i\cdot k\cdot{}^{m+k}C_k \\ &= j\cdot(m+k)\cdot{}^{m+k-1}C_{m-1} + i\cdot(m+k)\cdot{}^{m+k-1}C_{k-1} \\ &= n\cdot(j\cdot{}^{n-1}C_{m-1} + i\cdot{}^{n-1}C_{k-1}) \quad \text{is evidently divisible by } n . \end{aligned}$$

This is H.W. Lenstra's proof.

2nd Proof: ${}^nC_k$ is the number of n-bit strings with k bits set to 1 and $n-k$ bits 0 . Put these strings into boxes ("equivalence classes") as follows: Into each box put just those strings each obtainable from any other in the box by fewer than n one-bit circular shifts. For example, if $n = 5$ and $k = 3$ one box will hold the strings `1o1o1`, `o1o11`, `1o11o`, `o11o1` and `11o1o` . In general each box must hold n strings because no two circularly shifted versions of the same string can match; otherwise the number of one-bit circular shifts required to turn this string into a copy of itself would have to divide both n and k , and this is ruled out by $GCD(n, k) = 1$. Therefore the number ${}^nC_k$ of strings is the product of n by the number of boxes. This is Charles Holton's proof.

3rd Proof: ${}^nC_k = n\cdot{}^{n-1}C_k/(n-k)$ is an integer. $GCD(n, n-k) = GCD(n, k) = 1$, so ${}^{n-1}C_k/(n-k)$ must be an integer too, so n must divide ${}^nC_k$. This proof makes Problem 1 look too easy.

This document was created with FrameMaker 4.0.4

4th Proof: To show that ${}^nC_k/n = (n–1)\cdot(n–2)\cdot(\ldots)\cdot(n–k+1)/k!$ is an integer we must show for every prime p and integer exponent $K > 0$ that whenever p^K divides the denominator $k!$ it divides the numerator too. The exponent K of this prime p in the prime factorization of denominator $k!$ is

$$K := \lfloor k/p \rfloor + \lfloor k/p^2 \rfloor + \lfloor k/p^3 \rfloor + \ldots + \lfloor k/p^K \rfloor + \lfloor k/p^{K+1} \rfloor + \ldots$$

because each term $\lfloor k/p^m \rfloor$ in the series is the number of integers among $\{1, 2, 3, \ldots, k\}$ that is divisible by p^m . (This series contains only finitely many positive terms beyond the last of which all terms vanish.) The numerator $(n–1)\cdot(n–2)\cdot(\ldots)\cdot(n–k+1)$ is a product of $k–1$ consecutive integer factors among which the number divisible by p^m is *at least* $\lfloor (k–1)/p^m \rfloor$ because the first and last factors can *both* be divisible by p^m ; therefore the exponent of p in the numerator's prime factorization is at least

$$L := \lfloor (k–1)/p \rfloor + \lfloor (k–1)/p^2 \rfloor + \lfloor (k–1)/p^3 \rfloor + \ldots \lfloor (k–1)/p^L \rfloor + \lfloor (k–1)/p^{L+1} \rfloor + \ldots .$$

Now we must consider two cases according as prime p divides k or doesn't:

• If p does not divide k then every $\lfloor (k–1)/p^m \rfloor = \lfloor k/p^m \rfloor$ and therefore $L = K$, which implies that the largest p^K that divides the denominator $k!$ divides the numerator too, as claimed.

• If p divides k then p cannot divide n since $\mathrm{GCD}(n, k) = 1$. Now the largest p^K that divides the denominator $k!$ divides the numerator $(n–1)\cdot(n–2)\cdot(\ldots)\cdot(n–k+1)$ too because ${}^nC_k = n\cdot\big((n–1)\cdot(n–2)\cdot(\ldots)\cdot(n–k+1)\big)/k!$ is an integer.

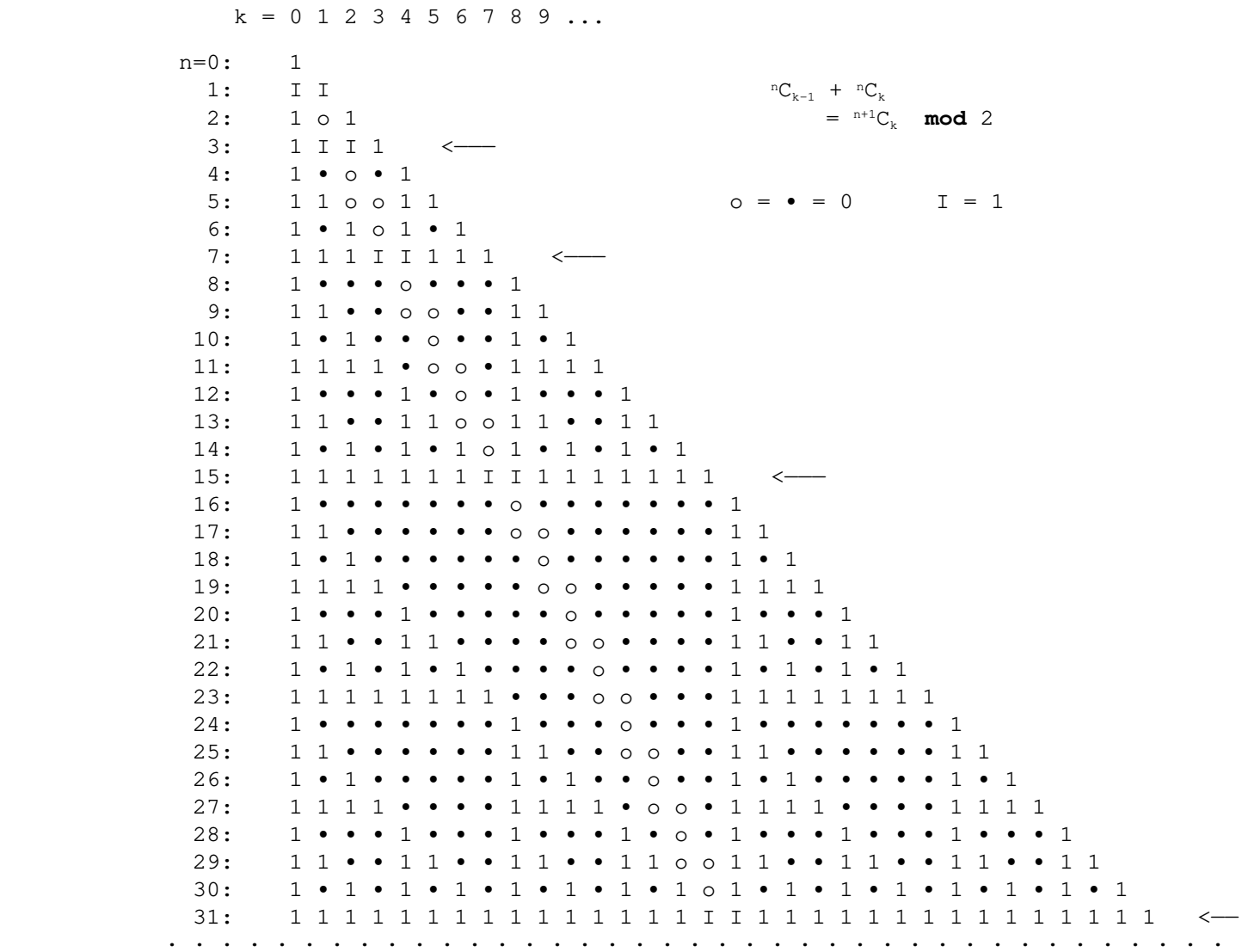
Therefore ${}^nC_k/n = (n–1)\cdot(n–2)\cdot(\ldots)\cdot(n–k+1)/k!$ is an integer as claimed. End of proof.

2 : For $k = 1, 2, 3, 4, \ldots$ prove that ${}^{2k–1}C_k$ is an odd integer just when $k = 2^K$ is a power of 2 .

Proof: The simplest proof is a pictorial proof, due to Prof. Raimund Seidel, based upon Pascal's recurrence ${}^{n+1}C_k = {}^nC_k + {}^nC_{k–1}$ in arithmetic **mod** 2 , which means $0+0 = 1+1 = 0$ and $1+0 = 0+1 = 1$. The picture is shown on the next page. In this picture the odd values of ${}^nC_k$ are plotted as 1's (or I's when $n = 2^{K+1} – 1$ and $k = 2^K – 1$ or 2^K) and the even values as • 's (or o 's when $k = n/2$ or $(n\pm1)/2$). The idea behind the picture is that it is composed recursively of triangles within triangles within triangles … . If Δ is a triangle whose sides and bottom row are made up entirely of 1 's , and if its top vertex corresponds to $n = k = 0$ while its bottom corresponds to $n = 2^K – 1$, then three such triangles can be assembled into the next larger triangle of twice the height thus:

Between the lower two triangles all plotted values are even. So ends the proof-by-picture.

An algebraic proof will be presented after the picture; and a third proof will come about because this problem 2 is a special case of the next problem 3.



Second proof for problem 2 : Every positive integer m has a binary representation for which are defined … U(m) := the number of 1 's in the binary representation of m ; and
T(m) := the number of trailing 0 's in the binary representation of m
= the exponent of 2 in the prime factorization of m .

Now we shall prove that $T(^{2k-1}C_k) = U(k) - 1$: Suppose k = X1ooo…ooo in binary, whence k–1 = Xo111…111 . Consequently T(k) + U(k) = 1 + U(k–1) for all k > 0 . Moreover $k\cdot{}^{2k-1}C_k = (2k-1)!/((k-1)!)2 = 2\cdot(2k-1)\cdot{}^{2k-3}C_{k-1}$, so $T(k) + T(^{2k-1}C_k) = 1 + T(^{2k-3}C_{k-1})$.
Next eliminate T(k) by subtraction to deduce that

$T(^{2k-1}C_k) - U(k) = T(^{2k-3}C_{k-1}) - U(k-1)$
$= T(^{2k-5}C_{k-2}) - U(k-2)$ … after replacing k by k–1 .
$= \ldots$ … repeatedly diminish k by 1 .
$= T(^{2-1}C_1) - U(1) = 0 - 1$, … as claimed.

Only when $k = 2^K$ is a power of 2 is U(k) = 1 , and then $T(^{2k-1}C_k) = 0$, which means that $^{2k-1}C_k$ is odd; otherwise it's even. End of second proof.

3 : Prove that ${}^{n}C_{k}$ is an odd integer just when the binary representation of n is the logical OR of the binary representations of k and n–k .

Proof: To better exploit symmetry in this problem let m := n–k and let

$$H(m, k) := {}^{m+k}C_k = (m+k)!/(m!\cdot k!) = {}^{m+k}C_m = H(k, m) ,$$

so our task is now to prove that H(m, k) is odd just when the binary representation of m+k is the logical OR of the binary representations of k and m . To this end let us define, for all nonnegative integers m and k , the nonnegative integers

m‡k = k‡m := logical OR of the binary representations of m and k ; and
m•k = k•m := logical AND of the binary representations of m and k .

For example take m := 12 = `oo11oo` and k := 10 = `oo1o1o` in binary; then m+k = 22 = `o1o11o` but m‡k = 14= `oo111o` and m•k = 8 = `oo1ooo` . Note that m+k = (m‡k) + (m•k) ; this is true in general because the nonzero bits of m•k are the *carries* generated when m and k are added. Consequently m+k = m‡k just when m•k = 0 ; in other words, what we have to prove is that

$$H(m, k) \text{ is odd just when } m\bullet k = 0 . \qquad (\dagger)$$

This has been done with two rather different proofs.

The first proof is by induction on m+k using Pascal's recurrence in the symmetrical form

$$H(m, k) = H(m–1, k) + H(m, k–1) . \qquad (\text{Verify this yourself.})$$

To start, H(m, 0) = 1 is odd and m•0 = 0 , so the desired result (†) is confirmed along the edges of Pascal's triangle and in the first two rows corresponding to n = 0 and n = 1 above. In the middle of the third row (n = 2) we find H(1, 1) = 2 is even and 1•1 = 1 ≠ 0 , confirming the desired result in the third row too. Let our induction hypothesis be that the desired result (†) is true for H(m, k–1) and H(m–1, k) at some m > 0 and k > 0 , from which we shall infer that the desired result holds also for H(m, k) = H(m–1, k) + H(m, k–1) as follows.

Let M := T(m) and K := T(k) be the numbers of trailing zeros in the binary representations of m and k respectively. If M = K then m•k ≠ 0 and (m–1)•k = m•(k–1) . For example try M = K = 3 , in which case m = …`1ooo` and k = …`1ooo` so m•k = …`1ooo` ≠ 0 , but m–1 = …`o111` and k–1 = …`o111` so (m–1)•k = …`oooo` = m•(k–1) . Consequently when M = K the induction hypothesis implies that H(m–1, k) and H(m, k–1) have the same parity (even or odd) so their sum H(m, k) must be even, which confirms the desired result (†). Therefore we need consider only the case M ≠ K henceforth, and by symmetry we can suppose M > K .

When M > K we find (m–1)•k ≠ 0 and m•(k–1) = m•k . For example when M > K = 3 we find m = …`oooo` , k = …`1ooo` , (m–1)•k ≥ `o1111`•`o1ooo` = `o1ooo` ≠ 0 and m•(k–1) = …`oooo` = m•k . Consequently when M > K the induction hypothesis implies that H(m–1, k) is even so that H(m, k–1) has the same parity as has H(m, k–1) + H(m–1, k) = H(m, k) , which agrees with m•(k–1) = m•k and thus propagates the induction hypothesis to H(m, k) . End of first proof.

The second proof uses the two functions $\lfloor x \rfloor$:= floor(x) = (the greatest integer in x) , and $f(x) := x - \lfloor x \rfloor$ = (the fractional part of x) . For example $\lfloor 3.78 \rfloor = 3$ and $f(3.78) = 0.78$. We shall apply these functions only to nonnegative arguments. It is not hard to see why

$$\lfloor x+y \rfloor - \lfloor x \rfloor - \lfloor y \rfloor = \lfloor f(x)+f(y) \rfloor = \text{ either } 0 \text{ or } 1$$

for all nonnegative x and y since $0 \le f(x) < 1$ and $0 \le f(y) < 1$. Moreover

$$\lfloor f(x) + f(y) \rfloor = 0 \quad \text{just when} \quad f(x) + f(y) < 1 .$$

Now recall $T(N) :=$ (the exponent of 2 in the prime factorization of integer N) . For example $T(12) = 2$. Apparently, for any positive integer N ,

$$T(N!) = \lfloor N/2 \rfloor + \lfloor N/4 \rfloor + \lfloor N/8 \rfloor + \lfloor N/16 \rfloor + \ldots + \lfloor N/2^K \rfloor + \ldots$$

because $\lfloor N/2^K \rfloor$ is the count of the number of positive integers divisible by 2^K and no bigger than N . The series contains only finitely many positive terms because $\lfloor N/2^K \rfloor = 0$ for all sufficiently big integers K . Now, for all positive integers m and k ,

$$\begin{aligned} T(H(m, k)) &= T((m+k)!) - T(m!) - T(k!) \\ &= \quad (\lfloor m/2 + k/2 \rfloor - \lfloor m/2 \rfloor - \lfloor k/2 \rfloor) \\ &\quad + (\lfloor m/4 + k/4 \rfloor - \lfloor m/4 \rfloor - \lfloor k/4 \rfloor) \\ &\quad + (\lfloor m/8 + k/8 \rfloor - \lfloor m/8 \rfloor - \lfloor k/8 \rfloor) \\ &\quad + (\lfloor m/16 + k/16 \rfloor - \lfloor m/16 \rfloor - \lfloor k/16 \rfloor) + \ldots \\ &= \quad \lfloor f(m/2) + f(k/2) \rfloor \\ &\quad + \lfloor f(m/4) + f(k/4) \rfloor \\ &\quad + \lfloor f(m/8) + f(k/8) \rfloor \\ &\quad + \lfloor f(m/16) + f(k/16) \rfloor + \ldots \\ &\geq 0 . \end{aligned}$$

Therefore $T(H(m, k)) = 0$ just when every $\lfloor f(m/2^K) + f(k/2^K) \rfloor = 0$ for every $K = 1, 2, 3, \ldots$, and only then is $H(m, k)$ odd. Thus, our proof requires that we determine for which pairs (m, k) are all the expressions $\lfloor f(m/2^K) + f(k/2^K) \rfloor = 0$ instead of some being 1 . For this purpose we shall extend the logical AND operation to define $x \bullet y$ also for nonintegers x and y whose binary representations have at most finitely many binary 1 's after the binary point. Define

$$x \bullet y := ((2^K \cdot x) \bullet (2^K \cdot y)) / 2^K$$

whenever an integer K exists for which both $2^K \cdot x$ and $2^K \cdot y$ are nonnegative integers. (The restriction to " finitely many binary 1 's after the binary point " is a technicality that precludes the possibility that numerically equal but different binary representations `X.Xo111111…` and `X.X1oooooo…` might prevent the numerical values of x and y from defining $x \bullet y$ uniquely.)

Note that the binary representation of $f(m/2^K)$ is obtained from that of m by shifting its bits K places right past the binary point and discarding bits still standing left of the point. For example

$$f(19/2^3) = f(\texttt{o1oo11} / \texttt{o1ooo}) = f(\texttt{o1o.o11}) = \texttt{o.o11} = 3/8 .$$

Also recall that $\lfloor f(x) + f(y) \rfloor = 0$ just when $f(x) + f(y) < 1$.

Now, $\lfloor f(m/2) + f(k/2) \rfloor = 0$ just when $f(m/2) \bullet f(k/2) = 0$; and if this is true then $\lfloor f(m/4) + f(k/4) \rfloor = 0$ just when $f(m/4) \bullet f(k/4) = 0$; and if those are both true then $\lfloor f(m/8) + f(k/8) \rfloor = 0$ just when $f(m/8) \bullet f(k/8) = 0$; and if all three of those are true then $\lfloor f(m/16) + f(k/16) \rfloor = 0$ just when $f(m/16) \bullet f(k/16) = 0$; and so on. In short, all of $\lfloor f(m/2^K) + f(k/2^K) \rfloor = 0$ just when $m \bullet k = 0$, and therefore that is when $H(m, k)$ is odd. End of second proof.

. . .

The solution to problem 3 also solves problem 2 as follows: For positive integers k , ${}^{2k-1}C_k = H(k, k-1)$ is odd just when $k \bullet (k-1) = 0$, which occurs just when $k = 2^K$ is a power of 2 .

Epilog

Each of the proofs presented above has been found by at least one U.C. Berkeley undergraduate taking lower division Math. courses including Math. H90 . But in a class like Math. 55 only a tiny percentage of the students can be expected to find even one of the proofs, much less three, even though they require no knowledge beyond what has already been taken up in Math. 55. If these problems are so hard, why put them before students almost all of whom will be frustrated?

Part of a Math. professor's job is to identify students with extraordinary mathematical capacities and to cultivate their talents. These are the people upon whom all of us will depend to decrypt an enemy's secret messages, to reveal the molecular structure of antibiotics and help synthesize them, to help design crashworthy cars and aircraft, to enhance the reliability and capacity of communication networks, to help reconcile stability with liveliness in economic systems, … , and to advance our understanding of persistent mathematical problems. These are the students who can solve problems like the three above or reproduce their solutions after a casual reading. Whether they work on Computer Science or on most of the best of modern mathematics, they will use what they learn in Math. 55 every day.

Part of a Math. professor's job is to help students learn as much mathematics as they will need to perform competently jobs that require a little of it occasionally. Some of these students will become the helpers or the employers of those mentioned in the previous paragraph; some will take responsibility for the planning and management of pure water distribution and safe sewage disposal, for the control of electric power grids that survive the loss of a power station without blacking out a whole state, for helping design safe and economical housing and commercial structures, … . These students should be able to follow every step of the proofs above and then, perhaps, appreciate the strategy that motivated the steps even if it is not a strategy that would have come to the student's mind unaided.

A few students will wish devoutly to avoid reading these proofs or, despite diligent attempts to read them, will be unable to follow the proofs' steps, finding too many that jump too far or get forgotten somewhere between their description and their application. They're not for everybody.

Part of a Math. professor's job is to help students discover before too late whether their talents include mathematics or lie elsewhere. Many students whose mathematical capabilities are limited either by nature or by earlier exposure to an unfortunate education (it is hard at times to tell which of these has played the greater role) will none the less get jobs involving computers, so these students should learn what they can and need to learn about computers. But where?

A student's time at U.C. Berkeley is precious; use it to learn well ideas that will serve you well throughout your life, ideas each of which was earned by a great mind's lifetime of effort. Most people who use computers have to know very little about them, and that little is accidental and ephemeral, better learned from a manual or book or community college or state university than from a crowded U.C.B. Computer Science class full of very competitive hot-shots.

Each student has to discover where his talents lie, and what this world needs done that he can do well regardless of whether he likes it. We in the older generations have an obligation so to order affairs that people who do well what needs doing well are rewarded for doing that. We will never get it quite right, but neither have we gotten it entirely wrong; and some of us are able to change.