MACM 101, D2, 12/01/2007. Lecture 3.

Puzzle of the day:

On computers which have 32-bit architecture a memory address is stored as a 32-bit binary number. How many memory addresses can be referenced on such computers, assuming that the memory is in the units of one byte (8 bits)?

Solution: The number of possible 32-bit binary strings is 2^{32}, because in each of 32 bits either 0 or 1 can go, so the number is $\underbrace{2 * 2 * \cdots * 2}_{32}$. Now, one kilobyte is $1024 = 2^{10}$ bytes, one megabyte is $1024 = 2^{10}$ kilobytes, and one gigabyte is 2^{10} megabytes, respectively (in computer science, a "round number" for 1000 is 1024, since 1024 is a power of 2). Now, $2^32 = 2^{10} * 2^{10} * 2^{10} * 4 = 4Gb$ of memory.

Combinations with repetition.

Definition 1. *An* alphabet *is a finite set of symbols, often called "letters". A (finite)* string *is a (finite) sequence of letters of alphabet. Finite strings are also called "words".*

How many 5-letter words (not necessarily existing words, just sequences of letters) out of 26 English letters? 26^5, since there are 26 possibilities for the first letter, times 26 possibilities for the second (the letters can repeat), and so on.

Formula: So the number of permutations with repetition of k letters over n-ary alphabet is n^k.

How many *combinations* of letters with repetition?

Example 1. *Suppose we want to express a number n as a sum of k non-negative numbers (0 allowed). How many such sums are there?*

For example, for $n = 7, k = 4$, $7 = 7+0+0+0, 6+1+0+0, 5+1+1+0, 5+2+0+0, 4+1+1+1, 4+2+1+0, 4+3+0+0, 3+2+1+1, 3+2+2+0, 3+3+1+0, 2+2+2+1$. The total is 11 possibilities.

Solution: *Express n in unary, and put "dividers" between summands. In this case, there are 3 dividers dividing 7 1s in 4 groups.* 1111—111——.

Assume for now that order matters, that is, $3+3+1+0$ and $0+3+1+3$ are different (later we will divide by $4!$ to account for this). Then the number of possibilities is the number of ways to put $k-1 = 3$ dividers in $n+k-1$ slots (choosing which slots are dividers, the rest are 1s).

Formula: The number of combinations with repetition of k elements out of n is equal to the number of ways to put dividers between elements of different kinds (or, alternatively, to put n elements in $n+k-1$ slots, leaving remaining slots for the dividers). There are k kinds of elements, and n elements total, so the formula is

$$\binom{n+k-1}{k-1} = \binom{n+k-1}{n}$$

In the book, the number of kinds of objects which we call k is n, and the number of objects we take which we call n is r.

In the example, we also ignore the order of elements, so for the case of splitting a sum the formula is $\frac{(n+k-1)!}{(k-1)!n!k!}$.

It is definitely possible for $k > n$: choosing more objects than there are kinds.

Example 2. *Suppose 40 students of MACM 101 go to Blenz to get coffee. Each of them gets one of the following: 1) coffee(c), 2) tea(t), 3) latte(l) 4) cappucino(p) 5) mocca(m). How many possible orders can there be, if they are ordering all drinks at the same time together?*

Solution: *Look at the list of drinks, ordered in alphabetical order and with dividers between different kinds. That is, an order is the string of the form*

$$cc \dots ccc|ll \dots lll|mmm \dots mm|pp \dots ppp|ttt \dots tt$$

There are 40 drinks and 5-1=4 dividers. In this case, all drinks are different, so we don't ignore the order of the groups. Applying the formula above, get $\binom{40+5-1}{4}$ *possible orders.*

Example 3. *Number of solutions of* $x_1 + x_2 + \dots x_n = r$ *is* $\binom{n+r-1}{r}$*. Proof idea: same as the number represented as a sum.*

What if the equation is $x_1 + x_2 + \dots x_n < r$*? Then consider an additional "slack variable* y*, and solve for the equation* $x_1 + x_2 + \dots x_n + y = r$*.*

Example 4. *How many times will the "print" statement execute in the following piece of code?*

for $i := 1$ **to** 20 **do**
  for $j := 1$ **to** i **do**
    for $k := 1$ **to** j **do**
      print$(i \cdot j + k)$

Solution: Can sum up the series, but too complicated. An easier way to solve this problem is to view each triple $< i, j, k >$ *as a triple of numbers where* $i \geq j \geq k$*. This is the same as considering all possible triples* $< i, j, k >$ *for* $1 \leq i, j, k \leq 20$*, but without order (so we are considering sets, not sequences). The number of triples of numbers between 1 and 20 without the order will be the number of ways to choose, with repetition, three numbers between 1 and 20 each. Therefore, the number of times the print statement is executed is* $\binom{3+20-1}{3} = 1540$.