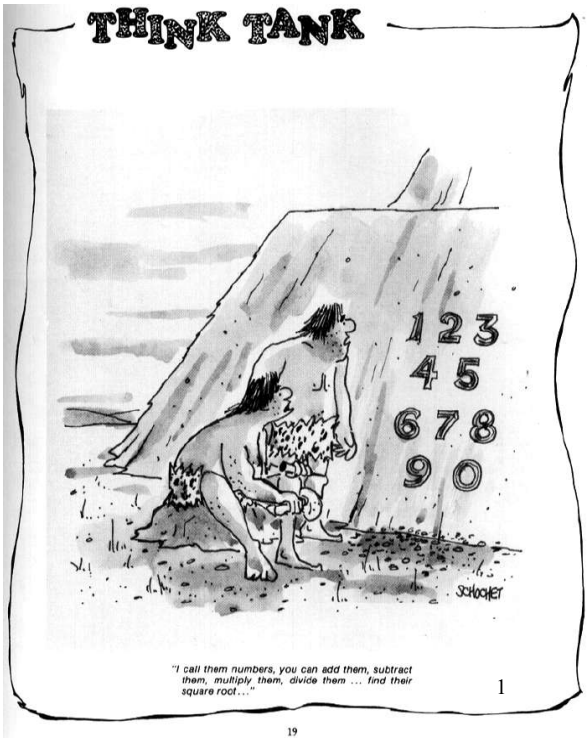
COMP 2121
DISCRETE MATHEMATICS

Lecture 3



Combinations

When dealing with any counting problem, we should ask ourselves about the importance of order in the problem. When order is relevant, we think in terms of permutations and arrangements and the rule of product. When order is not relevant, combinations could play a key role in solving the problem.

Example 1. (Review problem): How many 64 bit strings have exactly 6 1's?

[1] Reference: http://www.atariarchives.org/cartoon/showpage.php?page=19

Definition: *If we start with n distinct objects, each selection, or combination, or r of these objects, with no reference to order, corresponds to r! permutations of size r from the n objects. Thus the number of combinations of size r from collection of size n is*

$$C(n,r) = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!} = \binom{n}{r}$$ (read "*n choose r*".)

Example 2. a) Suppose that three employees are to be selected from among the ten available employes. In how many ways can this selection be made?

b) Suppose that three employees are to be selected from among the ten available to go to the same plant. In how many ways can this selection be made?

c) Suppose that three employees are to be selected from among the ten available to go to three different plants, one to each plant. In how many ways can this selection be made?

Example 3. A network consists of 10 servers; in particular two subnets (NET1 and NET2) with 5 servers in each. In order to crash a network, an attacker needs to crash exactly 7 of 10 servers. In how many possible ways can attacker try to crash the network if:

a) There are no further restrictions?
b) Attacker must crash exactly three servers from NET1 and four servers from the NET2?
c) Attacker must crash at least three servers from the NET1?

Example 4. Alice forgot her password for a mail account… She remembers that the password was some arrangement of letters in CANADIAN and that it had no adjacent A's. How many passwords does she possibly need to try?

Example 5.

- Function Random(x, y) returns a random integer, which satisfies $x \le Random(x, y) \le y$
- Function swap(i, j, X) swaps elements X[i] and X[j] in the array X
- Temp is an integer array of size 27
- RandInt is an integer array of size 4

Consider the following programming segment written in *pseudocode.*

```
For k = 0 to 26 {
        Temp[k] = k
        }

For k = 0 to 3 {
        swap(k, Random(k, 26), Temp)
        RandInt[k] = Temp [k]
        }

Sort RandInt in increasing order
```

What is the probability that after the segment execution, elements of the array RandInt will be as following:

RandInt[0] =1, RandInt [1] =3, RandInt[2] =4, RandInt [3] =4?

RandInt[0] =6, RandInt[1] =9, RandInt[2] =8, RandInt[3] = 2?

RandInt[0] =3, RandInt[1] =5, RandInt[2] =7, RandInt[3] = 8?

RandInt[0] =6, RandInt[1] =9, RandInt[2] =8, RandInt[3] = 2 when Sort is omitted?