

1. *Introduce yourself...*
2. *10 computers are brought in for servicing (and machines are serviced one at a time). Of the 10 computers, 3 are PCs, 4 are Macs, 2 are Linux machines, and 1 is an Amiga. Assume that all computers of the same type are indistinct (i.e. all the PCs are indistinct, all the Macs are indistinct, etc.).*

- (a) *In how many ways can the computers be ordered for servicing?*

$$\frac{10!}{3! 4! 2! 1!} = \frac{10!}{4! 3! 2} = 12600$$

- (b) *In how many distinct ways can the computers be ordered if the first 5 machines serviced must include all 4 Macs?*

Number of ways to order the first 5 machines by divider method (4 Macs, 1 other computer),

$$\frac{5!}{4! 1!} = 5$$

Number of ways non-Mac computers can be ordered,

$$\frac{6!}{3! 2! 1!} = 60$$

Total ways,

$$5 * 60 = 300$$

- (c) *In how many distinct ways can the computers be ordered if 1 PC must be in the first three and 2 PCs must be in the last three computers serviced?*

Number of ways to order the first 3 machines by divider method (1 PC, 2 other computers),

$$\frac{3!}{2! 1!} = 3$$

Number of ways to order the last 3 machines by divider method (2 PCs, 1 other computer),

$$\frac{3}{1! 2!} = 3$$

Number of ways non-PC computers can be ordered,

$$\frac{7!}{4! 2! 1!} = 105$$

Total ways,

$$3 * 3 * 105 = 945$$

3. *You are planning out what courses you want to take for the next two years. You have 22 courses to schedule over 6 quarters. All the classes are distinct, and order of classes within a quarter doesn't matter. How many different course plans are possible:*

- (a) *if there are no restrictions? (For example, you could put all 22 courses in one quarter if you wanted.)*

Total ways,

$$6^{22} = 1.31622e17$$

- (b) *if you can only take at most 4 courses in any quarter?*

Assuming 24 total items, 22 courses and 2 indistinct non-courses grouped in 6 groups of 4. Total ways,

$$\frac{24!}{(4!)^6 2!} = 1.62334e15$$

4. *A substitution cipher is derived from orderings of the alphabet. How many ways can the 26 letters of the alphabet (21 constants and 5 vowels) be ordered if each letter appears exactly once and:*

- (a) *there are no other restrictions?*

$$26! = 4.03291e26$$

- (b) *all five vowels must be next to each other?*

Number of positions to insert block of vowels by divider method (21 consonants, 1 vowel block),

$$\frac{22!}{21! 1!} = 22$$

Number of ways to order consonants,

$$21! = 5.10909e19$$

Number of ways to order vowels,

$$5! = 120$$

Total ways,

$$22 * 21! * 5! = 1.3488e23$$

- (c) *no two vowels can be next to each other?*

Each vowel can be inserted into a slot between two neighboring consonants. (e.g. \_ b \_ c \_ d \_ f \_ g \_ h \_ j \_ k \_ l \_ m \_ n \_ p \_ q \_ r \_ s \_ t \_ v \_ w \_ x \_ y \_ z \_)

Number of ways to order consonants,

$$21! = 5.10909e19$$

Number of options for choosing slots by combination,

$$\binom{22}{5} = \frac{22!}{5! 17!} = 26334$$

Number of ways to order vowels,

$$5! = 120$$

Total ways,

$$\frac{21! 22! 5!}{5! 17!} = \frac{22! 21!}{17!} = 1.61451e26$$

5. *You are counting cards in a card game that uses two standard decks of cards. There are 104 cards total. Each deck has 52 cards (13 values each with 4 suits). Cards are only distinct based on their suit and value, not which deck they came from.*

- (a) *In how many distinct ways can the cards be ordered?*

$$\frac{104!}{(2!)^{52}} = 2.28684e150$$

- (b) You are dealt two cards. How many distinct pairs of cards can you be dealt? Note: The order of the two cards does not matter.

$$\frac{\binom{104}{2}}{2!} = \frac{104!}{2! 102! 2!} = \frac{104 * 103}{4} = 2678$$

- (c) You are dealt two cards. Cards with values 10, jack, queen, king, and ace are considered "good" cards. How many ways can you get two "good" cards? Order does not matter.

There is a 10, jack, queen, king, and ace for each suit and 2 decks.

$$\frac{\binom{5*4*2}{2}}{2!} = \frac{40!}{2! 38! 2!} = \frac{40 * 39}{4} = 390$$

6. Imagine you have a robot that lives on a  $n \times m$  grid ( $n$  rows and  $m$  columns). The robot starts in cell  $(1, 1)$  and can take steps either to the right or down (no left or up steps). How many distinct paths can the robot take to the destination in cell  $(n, m)$ :

- (a) if there are no additional constraints?

Robot needs to take  $(m - 1)$  steps right and  $(n - 1)$  steps down in any order. Steps in the same direction are indistinct.

$$\frac{((m - 1) + (n - 1))!}{(m - 1)! (n - 1)!} = \frac{(m + n - 2)!}{(m - 1)! (n - 1)!}$$

, where  $m \geq 1$  and  $n \geq 1$ .

- (b) if the robot must start by moving to the right?

$$\frac{((m - 2) + (n - 1))!}{(m - 2)! (n - 1)!} = \frac{(m + n - 3)!}{(m - 2)! (n - 1)!}$$

, where  $m \geq 2$  and  $n \geq 1$ .

- (c) if the robot changes direction exactly 3 times? Moving down two times in a row is not changing directions, but switching from moving down to moving right is. For example, moving [down, right, right, down] would count as having two direction changes.

Consider two cases. There is one case for an initial move to the right, and one case for an initial move down. For three changes of directions, the robot must transition from the initial direction to the other direction and back. There must be two transitions from the initial direction of travel and one transition from the other direction of travel. Imagine there are slots between each step for which to choose transitions as shown below.

Case Start Right:  $r_1 \_ \dots r_{m-1} \_ t \mid d_1 \_ \dots d_{n-1}$ ,  $t$  is a required transition

Case Start Down:  $r_1 \_ \dots r_{m-1} \mid d_1 \_ \dots d_{n-1} \_ t$ ,  $t$  is a required transition

For each case, there are  $m - 2$  options for choosing the first transition from moving right and  $n - 2$  options for choosing the first transition from moving down. One choice is made for each first transition for each case.

$$\binom{m-2}{1} * \binom{n-2}{1} * 2 = (m-2) * (n-2) * 2$$

, where  $m \geq 3$  and  $n \geq 3$ .

7. Consider an array  $x$  of integers with  $k$  elements (e.g.  $\text{int } x[k]$ ), where each entry in the array has a distinct integer value between 1 and  $n$ , inclusive, and the array is sorted, so  $x[0] < x[1] < \dots < x[k-1]$ .

*How many such sorted arrays are possible?*

There is only one such array for each case.

8. *Given all the start-up activity going on in high-tech, you realize that applying combinatorics to investment strategies might be an interesting idea to pursue. Say you have \$20 million that must be invested among 4 possible companies. Each investment must be in integral units of \$1 million, and there are minimal investments that need to be made if one is to invest in these companies. The minimal investments are \$1, 2, 3, and 4 million for company 1, 2, 3, and 4, respectively. How many different strategies are available if*

- (a) *an investment must be made in each company?*

Number of ways to divide money with no restrictions by divider method (20 \$1 million chunks, 3 dividers),

$$\frac{(20 + 3)!}{20! 3!} = 1771$$

Number of ways to divide money such that Company 1 gets 0 chunks by divider method (20 \$1 million chunks, 2 dividers),

$$\frac{(20 + 2)!}{20! 2!} = 231$$

Number of ways to divide money such that Company 2 gets 0 chunks and Company 1 has at least 1 chunk by divider method (19 \$1 million chunks, 2 dividers),

$$\frac{(19 + 2)!}{19! 2!} = 210$$

Number of ways to divide money such that Company 2 gets 1 chunk and Company 1 has at least 1 chunk by divider method (18 \$1 million chunks, 2 dividers),

$$\frac{(18 + 2)!}{18! 2!} = 190$$

Number of ways to divide money such that Company 3 gets 0 chunks, Company 1 has at least 1 chunk, and Company 2 has at least 2 chunks by divider method (17 \$1 million chunks, 2 dividers),

$$\frac{(17 + 2)!}{17! 2!} = 171$$

Number of ways to divide money such that Company 3 gets 1 chunk, Company 1 has at least 1 chunk, and Company 2 has at least 2 chunks by divider method (16 \$1 million chunks, 2 dividers),

$$\frac{(16 + 2)!}{16! 2!} = 153$$

Number of ways to divide money such that Company 3 gets 2 chunks, Company 1 has at least 1 chunk, and Company 2 has at least 2 chunks by divider method (15 \$1 million chunks, 2 dividers),

$$\frac{(15 + 2)!}{15! 2!} = 136$$

Number of ways to divide money such that Company 4 gets 0 chunks, Company 1 has at least 1 chunk, Company 2 has at least 2 chunks, and Company 3 has at least 3 chunks by divider method (14 \$1 million chunks, 2 dividers),

$$\frac{(14 + 2)!}{14! 2!} = 120$$

Number of ways to divide money such that Company 4 gets 1 chunk, Company 1 has at least 1 chunk, Company 2 has at least 2 chunks, and Company 3 has at least 3 chunks by divider method (13 \$1 million chunks, 2 dividers),

$$\frac{(13+2)!}{13!2!} = 105$$

Number of ways to divide money such that Company 4 gets 2 chunks, Company 1 has at least 1 chunk, Company 2 has at least 2 chunks, and Company 3 has at least 3 chunks by divider method (12 \$1 million chunks, 2 dividers),

$$\frac{(12+2)!}{12!2!} = 91$$

Number of ways to divide money such that Company 4 gets 3 chunks, Company 1 has at least 1 chunk, Company 2 has at least 2 chunks, and Company 3 has at least 3 chunks by divider method (11 \$1 million chunks, 2 dividers),

$$\frac{(11+2)!}{11!2!} = 78$$

Number of ways to divide money such that the requirements are met,

$$1771 - 231 - 210 - 190 - 171 - 153 - 136 - 120 - 105 - 91 - 78 = 286$$

(b) *investments must be made in at least 3 of the 4 companies?*

Number of ways to divide money such that Company 1 and 2 get 0 chunks, Company 3 has at least 3 chunks, and Company 4 has at least 4 chunks (13 \$1 million chunks, 1 divider),

$$\frac{(13+1)!}{13!1!} = 14$$

Number of ways to divide money such that Company 1 and 3 get 0 chunks, Company 2 has at least 2 chunks, and Company 4 has at least 4 chunks (13 \$1 million chunks, 1 divider),

$$\frac{(14+1)!}{14!1!} = 15$$

Number of ways to divide money such that Company 1 and 4 get 0 chunks, Company 2 has at least 2 chunks, and Company 3 has at least 3 chunks (15 \$1 million chunks, 1 divider),

$$\frac{(15+1)!}{15!1!} = 16$$

Number of ways to divide money such that Company 2 and 3 get 0 chunks, Company 1 has at least 1 chunk, and Company 4 has at least 4 chunks (15 \$1 million chunks, 1 divider),

$$\frac{(15+1)!}{15!1!} = 16$$

Number of ways to divide money such that Company 2 and 4 get 0 chunks, Company 1 has at least 1 chunk, and Company 3 has at least 3 chunks (16 \$1 million chunks, 1 divider),

$$\frac{(16+1)!}{16!1!} = 17$$

Number of ways to divide money such that Company 3 and 4 get 0 chunks, Company 1 has at least 1 chunk, and Company 2 has at least 2 chunks (17 \$1 million chunks, 1 divider),

$$\frac{(17+1)!}{17!1!} = 18$$

Number of ways to divide money such that 3 or more companies get 0 chunks,

$$\binom{4}{3} + 1 = \frac{4!}{3!1!} + 1 = 5$$

Number of ways to divide money such that the requirements are met,

$$1771 - 14 - 15 - 16 - 16 - 17 - 18 - 5 - 190 - 153 - 136 - 105 - 91 - 78 = 917$$

9. Say we send out a total of 26 distinct emails to 10 distinct users where each email we send is equally likely to go to any of the 10 users. (Note that it is possible that some users may not actually receive any mail from us.) What is the probability that the 26 emails are distributed such that there are 4 users who receive exactly 2 emails each and 3 users who receive exactly 6 each?

Number of possible outcomes,

$$|S| = 10^{26}$$

Number of ways that 4 users can receive 2 emails each and 3 users can receive 6 emails each can be determined by finding the number of combinations for choosing 4 users then 3 users from the group of 10. Only these 7 users will receive emails since the emails total 26.

$$|E| = \binom{10}{4} * \binom{6}{3} = \frac{10!}{4!6!} * \frac{6!}{3!3!} = 4200$$

$$P(E) = \frac{|E|}{|S|} = 4.2e-23$$

10. To get good performance when working binary search trees (BST), we must consider the probability of producing completely degenerate BSTs (where each node in the BST has at most one child).
- (a) If the integers 1 through  $n$  are inserted in arbitrary order into a BST (where each possible order is likely), what is the probability (as an expression in terms of  $n$ ) that the resulting BST will have completely degenerate structure?

To make a degenerate BST (DBST), the greatest or least number remaining in the set must be chosen each iteration. There are 2 options for choosing a number each iteration and  $n - 1$  iterations.

$$|S| = n!, |E| = 2^{n-1} \implies P(E) = \frac{2^{n-1}}{n!}$$

- (b) Using your expression from part (a), determine the smallest value of  $n$  for which the probability of forming a completely degenerate BST is less than .001 (0.1 %)

$$n = 9$$

11. Say a hacker has a list of  $n$  distinct password candidates, only one of which will successfully log her into a secure system.
- (a) If she tries passwords from the list at random, deleting those passwords that do not work, what is the probability that her first successful login will be (exactly) on her  $k$ -th try?

Each time a password is chosen, the probability that the next choice will be correct will increase because the used passwords are being deleted.

$$P(E) = \frac{1}{n - k}$$

- (b) Now say the hacker tries passwords from the list at random, but does not delete previously tried passwords from the list. She stops after her first successful login attempt. What is the probability that her first successful login will be (exactly) on her  $k$ -th try?

$$P(E) = \frac{1}{n}$$

12. Say a university is offering 3 programming classes: one in Java, one in C++, and one in Python. The classes are open to any of the 100 students in the university. There are:

- a total of 32 students in the Java class
- a total of 24 students in the C++ class
- a total of 21 students in the Python class
- 12 students in both the Java and C++ classes (Note: These students are also counted as being in each class in the numbers above.)
- 10 students in both the Java and Python classes
- 3 students in all 3 classes (Note: These students are also counted as being in each pair of classes in the numbers above.)

- (a) If a student is chosen randomly at the university, what is the probability that the student is not in any 3 of the programming classes?

$E_j$  is the set of students taking the Java course,  $E_c$  is the set of students taking the C++ course, and  $E_p$  is the set of students taking the Python course.

$$|E_j \cup E_c \cup E_p| = |E_j + E_c + E_p - E_j \cap E_c - E_j \cap E_p - E_c \cap E_p + E_j \cap E_c \cap E_p| = 51$$

$$P(x \notin E_j \cup E_c \cup E_p) = \frac{49}{100} = .49$$

- (b) If a student is chosen randomly at the university, what is the probability that the student is taking exactly one of the three programming classes?

$$|E_j \cap E_c - E_j \cap E_c \cap E_p| = 9$$

$$|E_j \cap E_p - E_j \cap E_c \cap E_p| = 7$$

$$|E_c \cap E_p - E_j \cap E_c \cap E_p| = 4$$

$$|E_j \cap E_c \cap E_p| = 3$$

$$P(x \text{ in exactly 1 class}) = \frac{51 - (9 + 7 + 4 + 3)}{100} = .28$$

- (c) If two different students are chosen randomly at the university, what is the probability that at least one of the chosen students is taking at least one of the programming classes?

$$P(x \text{ or } y \notin E_j \cup E_c \cup E_p) = \frac{49}{100} * \frac{49}{99} = .24$$

$$P(x \text{ or } y \in E_j \cup E_c \cup E_p) = 1 - P(x \text{ or } y \notin E_j \cup E_c \cup E_p) = .76$$

13. A binary string containing  $m$  0s and  $n$  1s (in arbitrary order, where all orderings are equally likely) is sent over a network. What is the probability that the first  $r$  bits of the received message contain exactly  $k$  1s?

Number of ways to get  $k$  1s,

$$\frac{r!}{k!(r-k)!}$$

$$P(\text{bit} = 1) = \frac{n}{m+n}$$

$$P(k \text{ bits} = 1) = \frac{r!}{k!(r-k)!} * \left(\frac{n}{m+n}\right)^k * \left(\frac{m}{m+n}\right)^{r-k}$$

, where  $0 \leq k \leq r$ .

14. A computer generates two random integers in the range 1 to 12, inclusive, where each value in the range 1 to 12 is equally likely to be generated.

- (a) What is the probability that the second randomly generated integer has a value that is strictly greater than the first?

$$P(x_1 = 1) = P(x_1) = 2 = \dots = P(x_1) = \frac{1}{12} = .083$$

$$P(x_2 > x_1) = \frac{12 - x_1}{12}$$

$$\sum_{x_1=1}^{12} \frac{1}{12} * \frac{12 - x_1}{12} = \frac{66}{144} = .4583$$

- (b) **[coding]** Write a function in the programming language of your choice that takes in a number of trials, runs that many simulations of this experiment, and returns the fraction of the trials in which the second integer is greater.

Program returned 0.45868