Binary Sequences

Written by Rachel J. Morris, last updated Jan 16, 2018

About

In this section we will learn about Binary Sequences, and how they can be used to model problems.

Topics

1. Binary Sequences
2. Unordered Lists

Binary Sequences

1. Binary Sequences

Question: How many binary sequences can we build with one 0 and two 1's?

Notes

1. Binary Sequences

Question: How many binary sequences can we build with one 0 and two 1's?

1) **011**
2) **101**
3) **110**

For a short enough sequence, we can write out all possibilities. But this isn't effective for longer strings.

Notes

1. Binary Sequences

The Binary Sequence Theorem

"The number of binary sequences with r 1's and n – r 0's is C(n, r) or C(n, n – r). a"

From Discrete Mathematics, Ensley and Crawley, page 409

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

The Binary Sequence Theorem

"The number of binary sequences with r 1's and n – r 0's is C(n, r) or C(n, n – r). a"

From Discrete Mathematics, Ensley and Crawley, page 409

So if we have one 0 and two 1's, we can solve it like this:

- Amount of 1's: r = 2
- Amount of 0's: n – r = 1

- Solve for *n*: n – 2 = 1 n = 1 + 2 **n = 3**
- Solve: **C(3, 2) = 3**

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

We can use this for any arbitrary number of 1's and 0's.

Example: How many binary sequences are there with five 1's and three 0's?

- What is ***r*?**
- What is ***n*?**
- What is ***C(n, r)*?**

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

We can use this for any arbitrary number of 1's and 0's.

Example: How many binary sequences are there with five 1's and three 0's?

- What is ***r*?** $r = 5$

- What is ***n*?** $n - r = 3$; $n - 5 = 3$; $n = 8$

- What is ***C(n, r)*?** $C(8, 5) = 56$

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

We can generalize this idea into problems where we need to find ***"r"*** of something, and ***"n-r"*** of something else. We can use the 1's and 0's to represent pieces of data.

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

We can't use methods from before because we don't know how many A's, B's, and C's will be used!

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

Instead, think of these like groupings:

- *a* amount of A's in the first group
- *b* amount of B's in the second group
- *c* amount of C's in the third group

Then, we also need to think of how they're ***separated***, and this also becomes something that we count.

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

"AA + separator + BB + separator + C"

Because we have 3 letters to choose from, we need 2 separators. It will always be one less than the amount of options.

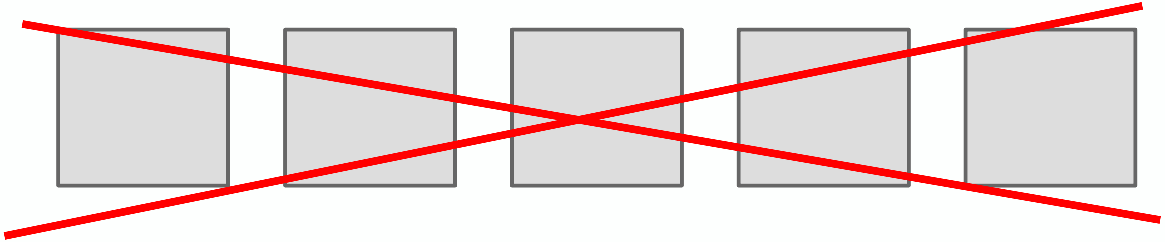
Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

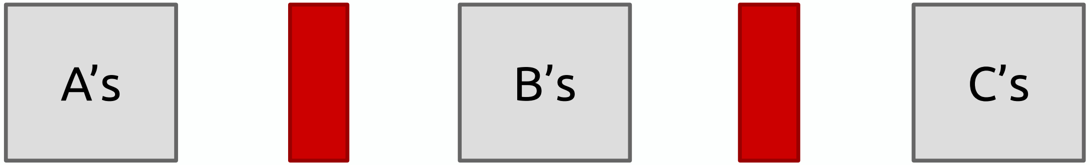
1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

So instead of thinking of the problem as 5 letters to be filled in…



We think of it as the A-group, a separator, the B-group, a separator, and the C-group.



Notes

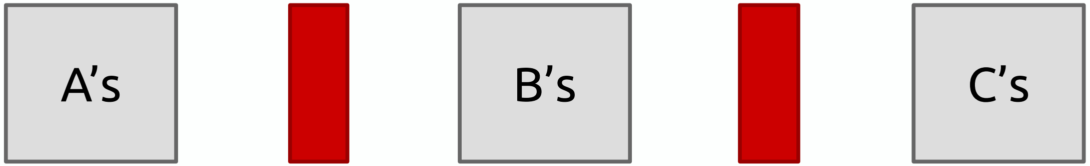
The # of binary sequences with ***r*** 1’s and ***n – r*** 0’s is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

We have 5 spaces for letters, but now we also have to count two separators. Total, there are 7 “spaces” to fill.

5 letters, and 2 separators = 7 total spaces.



Notes

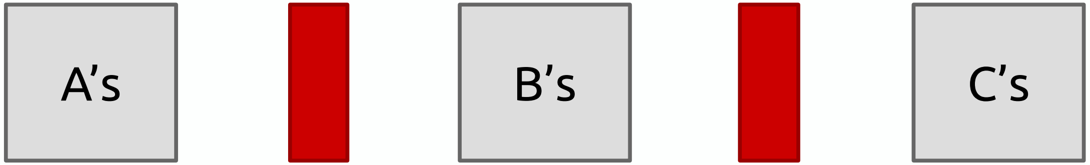
The # of binary sequences with ***r*** 1’s and ***n – r*** 0’s is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

5 letters, and 2 separators = 7 total spaces.

We can represent the letters with the digit “0”, and the separators with the digit “1”. We don’t really care where A, B, or C goes, just that we need to fill 5 letters.



Notes

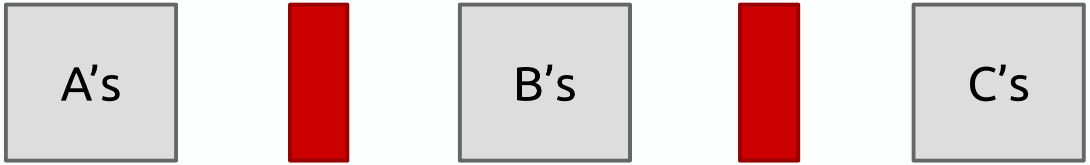
The # of binary sequences with ***r*** 1’s and ***n – r*** 0’s is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

5 letters = five 0’s; 2 separators = two 1’s

Now that we’re modeling the problem with 0’s and 1’s, we can use the Binary Sequences theorem to solve it.



Notes

The # of binary sequences with ***r*** 1’s and ***n – r*** 0’s is ***C(n, r)***.

1. Binary Sequences

Example: How many 5-letter strings can we build with only the letters {A, B, C}?

5 letters = five 0's; **2 separators = two 1's**

$$r = 5$$

$$n - r = 2; \quad n - 5 = 2; \quad n = 7$$

$$C(7, 5) = 21$$

Notes

The # of binary sequences with r 1's and $n - r$ 0's is $C(n, r)$.

Unordered Lists

2. Unordered Lists

We can further use the binary sequence model to solve problems dealing with unordered lists.

Remember that unordered lists are structures where **repetitions are allowed** and **order doesn't matter**.

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

2. Unordered Lists

Example: If you have a bag that can hold 10 pieces of fruit, and the store you're at sells only 3 types of fruit, how many ways can you fill the bag?

Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

2. Unordered Lists

Example: If you have a bag that can hold 10 pieces of fruit, and the store you're at sells only 3 types of fruit, how many ways can you fill the bag?

fruitA_count + fruitB_count + fruitC_count = 10

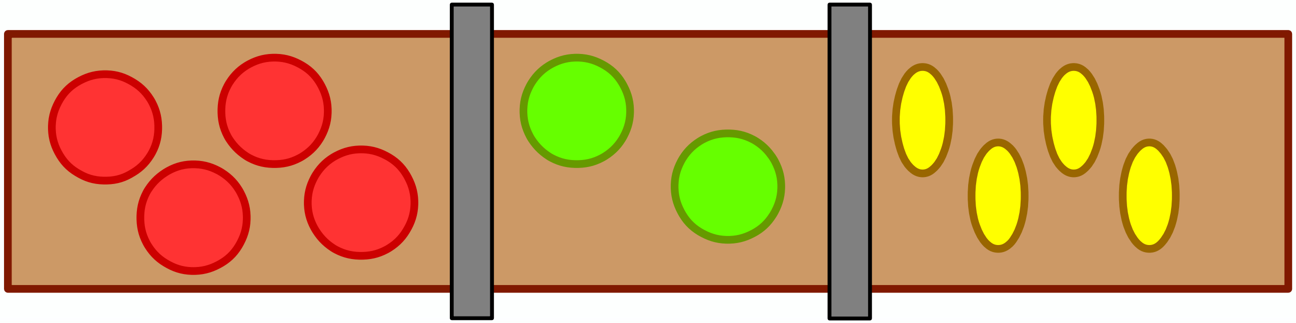
Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

2. Unordered Lists

Example: If you have a bag that can hold 10 pieces of fruit, and the store you're at sells only 3 types of fruit, how many ways can you fill the bag?

In this case, we again generalize it so that the fruits are all represented with 0's and the **separators** are represented with 1's.



Notes

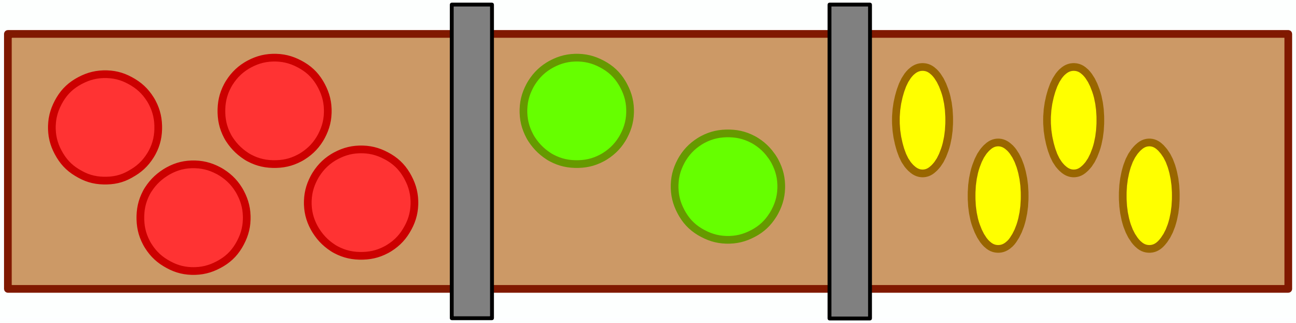
The # of binary sequences with ***r*** 1's and ***n* – *r*** 0's is ***C(n, r)***.

2. Unordered Lists

Example: If you have a bag that can hold 10 pieces of fruit, and the store you're at sells only 3 types of fruit, how many ways can you fill the bag?

Fruits are 0's, so we can say $r = 10$

Separators are 1's, so we can say $n - r = 2$; $n - 10 = 2$; $n = 12$.



Notes

The # of binary sequences with ***r*** 1's and ***n – r*** 0's is ***C(n, r)***.

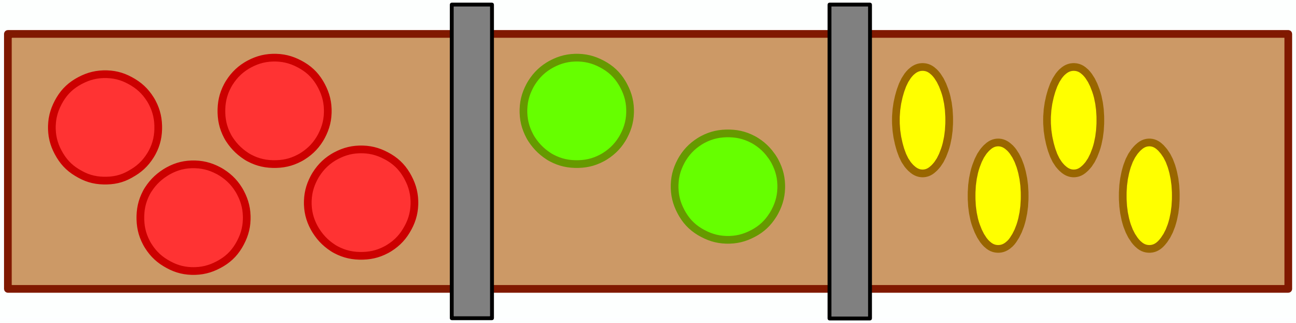
2. Unordered Lists

Example: If you have a bag that can hold 10 pieces of fruit, and the store you're at sells only 3 types of fruit, how many ways can you fill the bag?

C(12, 10) = 66.

Note that we can also say separators are 0's and fruits are 1's; it doesn't matter. It will come out to the same thing.

$r = 2$; $n - r = 10$; $n - 2 = 10$; $n = 12$; $C(12, 2) = 66$



Notes

The # of binary sequences with ***r*** 1's and ***n* – *r*** 0's is ***C(n, r)***.

Conclusion

Binary Sequences are one of the more difficult structures in this chapter to “decipher” from the word problem to an actual formula. It just takes some practice. Once you’re able to recognize that a problem is an unordered list, you will have the formula to solve the problem.