MERRIMACK COLLEGE

CSC 6000 Week 5

Combinatorics and Binomial Coefficients

Basic Programming Concepts and Discrete Mathematics - Dr. Paulo Fernandes

Presentation Agenda

Week 5

- Combinatorics
  a. Permutations, Combinations, and Partitions
- Combinations
  a. Basic Combinations
     i. Product of a sequence
  b. Binomial Coefficients
     i. Applying combinations
  c. Multicombinations
- Partitions
- This Week's tasks

MERRIMACK COLLEGE

Combinatorics

Previously on combinatorics …

Just a permutation!

Arrangements

Counting is natural, as we count the days, the weeks (this is the fourth by the way), months, years, choices, possibilities, etc.

Combinatorics is an analysis tool to assess possible outcomes, but also to define structures.

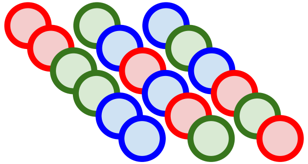
The main kinds of combinatorics are enumerative and are applied to compute:

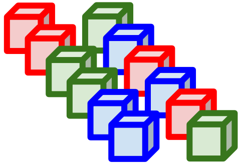
- permutations,
- combinations, and
- partitions.

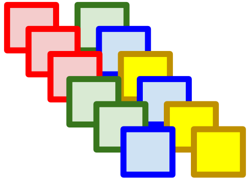
Enumerative Combinatorics

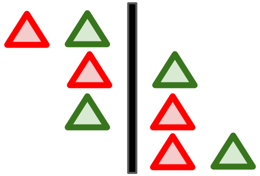
The usual problems solved by enumerative combinatorics are:

- Permutations
  - How many ways to sort the elements of a set
- Arrangements
  - How many ways to sort some of the elements of a set
- Combinations
  - How many subsets of a given size from a set
- Partitions
  - How many ways to split the elements of a set









MERRIMACK COLLEGE

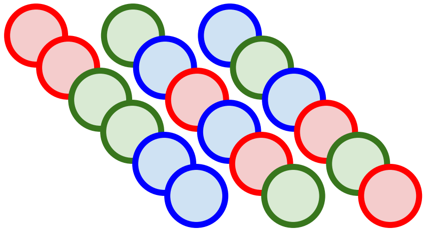
Wikipedia: Combinatorics.

Enumerative Combinatorics

Permutations

- How many ways to sort the elements of a set

We saw last week how to compute these in a few variations.



From a practical point of view, these are the most common basic problems in combinatorics. Permutations can be applied to answer questions like:

- Three athletes, **Joe**, **Jack**, and **Jeff** are competing in a final round. How many outcomes are possible for the podium?

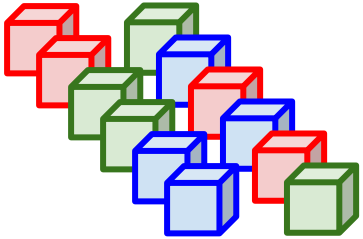
Wikipedia: Permutation.

Enumerative Combinatorics

Arrangements

- How many ways to sort some of the elements of a set

We saw last week how to compute these in a few variations.



From a practical point of view these are a variation of permutations. These can be applied to answer questions like:

- You bought three boxes, **A**, **B**, and **C**. Now you need to pick two of them - one for the kitchen, and one for the garage. How many choices do you have?

Wikipedia: Permutation.

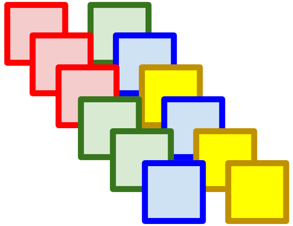
Enumerative Combinatorics

Combinations

- How many subsets of a given size from a set

We will see today how to compute these in a few variations.

From a practical point of view, these are the oldest problems in combinatorics. Combinations can be applied to answer questions like:



- Four state teams, **MA**, **NY**, **FL**, and **CA** are competing in a pool. How many matches are needed for all teams to play against each other?

Wikipedia: Combination.

Enumerative Combinatorics

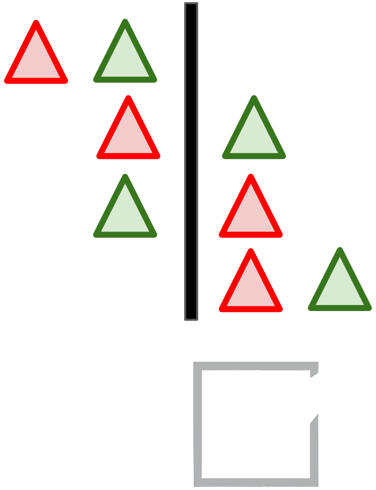
Partitions

- How many ways to split the elements of a set

We will see today how to compute it in a few variations.

This is one of the more common problems in combinatorics. Partitions can be applied to answer questions like:

- Two sisters, **Jane** and **Joan**, are entering into a competition where the contenders are split into two different pools. In how many ways the sisters can be placed in the two available pools?



Wikipedia: Partition of a set.

Combinations

Life is full of permutations and combinations. Sometimes the order you do things matters, sometimes it doesn't, but in order to find the solution in life you must work through each possibility presented to find your opportunity.
Gregory Willis

Given a set of elements, how many subsets of a given size can be generated?

And what would these subsets be?

As we will see, combinations can be computed considering elements which are unique - or not:

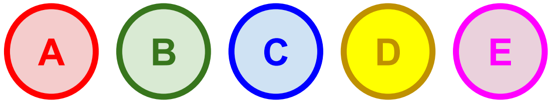
- **Basic combinations**
- Binomial Coefficients
- Multicombinations

Previously on combinatorics...

Arrangements

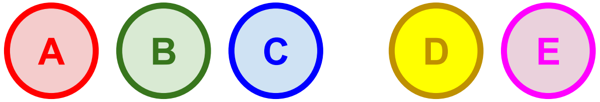
- All possible permutations for a set of ***n*** elements are ***n!***
- If we limit it to just the first ***k*** elements we are leaving out ***n-k*** elements and their permutations
  - the number of permutations of ***n-k*** elements is given by ***(n-k)!***
    - Thus, the arrangements of ***k*** elements out of ***n*** will be denoted by ***A(k, n)*** and computed as:

$$A(k,n) = \frac{n!}{(n-k)!}$$



P(5) = 5! = 120

Now considering not 5, but just 3 elements:



A(3,5) = 5! / 2! = 60

MERRIMACK COLLEGE

Calculator: K-Permutations.

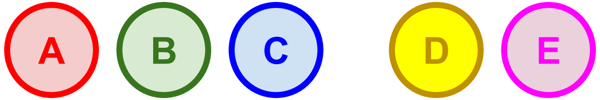
Basic Combinations

- The arrangements of ***k*** elements out of ***n*** is:

$$A(k,n) = \frac{n!}{(n-k)!}$$

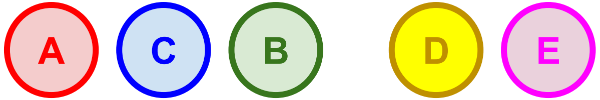
- The combination of ***k*** elements out of ***n*** is considered ignoring all permutations of those chosen k elements, as the order is not important. Thus the combination of ***k*** elements out of ***n*** is denoted by ***C(k,n)*** and is given by:

$$C(k,n) = \frac{A(k,n)}{k!} = \frac{n!}{(n-k)!k!}$$



A(3,5) = 5! / 2! = 60

Now ignoring the permutations of 3 elements:



C(3,5) = 5! / 2! 3! = 10

MERRIMACK COLLEGE

Calculator: Combinations.

From Summations to Products

- We have seen summations before, which is basically like a **for** loop where you keep on adding the terms, such as:

$$\sum_{i=1}^{5} i = 1 + 2 + 3 + 4 + 5 = 15$$

- We have a similar operator, which performs a product instead of adding the terms, such as:

$$\prod_{i=1}^{5} i = 1 * 2 * 3 * 4 * 5 = 5! = 120$$

$$\sum_{\text{variable} = \text{initial}}^{\text{final}} \text{term}$$

$$\prod_{\text{variable} = \text{initial}}^{\text{final}} \text{term}$$

Wikipedia: Product of a sequence.

Products of a Sequence

- The product of a sequence is denoted using the capital Greek letter pi, and it works just like a **for** loop where you keep on multiplying the terms - such as a factorial, but also others:

$$\prod_{\text{variable} = \text{initial}}^{\text{final}} \text{term}$$

$$\prod_{i=1}^{5} i = 1 * 2 * 3 * 4 * 5 = 5! = 120$$

$$\prod_{i=4}^{7} i = 4 * 5 * 6 * 7 = \frac{7!}{3!} = 840$$

$$\prod_{i=1}^{5} 2i = 2 * 4 * 6 * 8 * 10 = 3,840$$

```
def prod4to7():
    ans = 1
    for i in range(4,8):
        ans *= i
    return ans
print("Product 4 to 7:", prod4to7())
Product 4 to 7: 840
```

MERRIMACK COLLEGE

Video: Product Notation.

Combinations

- The combination of ***k*** elements out of ***n*** can be denoted by ***C(k,n)*** and is given by:

$$C(k,n) = \frac{n!}{(n-k)!k!}$$

- To implement the computation of ***C(k, n)*** a more compact notation is often used:

$$n^{\underline{k}} = n(n-1)\cdots(k+1)k = \prod_{i=k}^{n} i = \frac{n!}{(n-k)!}$$

$$C(k,n) = \frac{n^{\underline{k}}}{k!}$$

```python
def comb(k,n):
    ans = 1
    for i in range(n, n-k, -1):
        ans *= i
    for i in range(2, k+1):
        ans /= i
    return ans
```

```python
print("Combination of 2 over 5:", comb(2,5))
print("Combination of 3 over 6:", comb(3,6))
print("Combination of 4 over 8:", comb(4,8))
print("Combination of 5 over 12:", comb(5,12))
print("Combination of 6 over 10:", comb(6,10))
print("Combination of 7 over 12:", comb(7,12))
```

```
Combination of 2 over 5: 10.0
Combination of 3 over 6: 20.0
Combination of 4 over 8: 70.0
Combination of 5 over 12: 792.0
Combination of 6 over 10: 210.0
Combination of 7 over 12: 792.0
```


Text: Product Notation.

Combinations

Life is full of permutations and combinations. Sometimes the order you do things matters, sometimes it doesn’t, but in order to find the solution in life you must work through each possibility presented to find your opportunity.
Gregory Willis

Given a set of elements, how many subsets of a given size can be generated?

And what would these subsets be?

As we will see, combinations can be computed considering elements which are unique - or not:

- Basic combinations
- **Binomial Coefficients**
- Multicombinations

Combinations, a.k.a. Binomial Coefficients

- The combination of ***k*** elements out of ***n*** can be denoted by ***C(k,n)*** and is given by:

$$C(k,n) = \frac{n!}{(n-k)!k!}$$

- It can be said in many other ways, like ***n*** *choose* ***k***, *combination of* ***n*** *in* ***k***, ***k***-*combinations over* ***n***.
- Since it's a very old definition in mathematics, several mathematical traditions define it with a different notation, for example:

This last one is particularly accepted within the academic community and it is called a ***binomial coefficient***.

$$C(k,n) = C(n,k) = {}^{n}C_k = {}_{n}C_k = C^n_k = C^k_n = C_{n,k} = \binom{n}{k}$$

MERRIMACK COLLEGE

Wikipedia: Binomial Coefficients.

Binomial Coefficients

- The combination of ***k*** elements out of ***n*** can be rewritten as:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

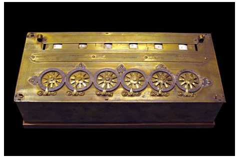
```python
def comb(k,n):
    ans = 1
    for i in range(k):
        ans *= (n-i) / (i+1)
    return ans
```

$$\binom{n}{k} = \frac{n^{\underline{k}}}{k!} = \frac{n(n-1)(n-2)\cdots(n-(k-1))}{k(k-1)(k-2)\cdots 1} = \prod_{i=1}^{k} \frac{n+1-i}{i};$$

Wikipedia: [Binomial Coefficients](#).

Binomial Coefficients Properties

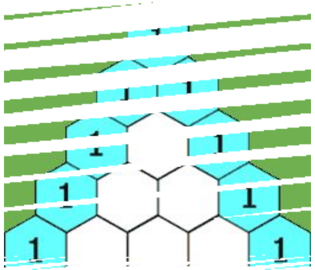
1623 - 1662



- Blaise Pascal, the French Philosopher, creator of the Pascaline calculator machine in 1642, was one of the early Western mathematicians who deeply studied binomial coefficients.
  - He defined *Pascal's triangle*, a way to dispose binomial coefficients and verify their properties.

$$
\begin{matrix}
 & & & & \binom{0}{0} & & & & \\
 & & & \binom{1}{0} & & \binom{1}{1} & & & \\
 & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & \\
 & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & \\
\binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4}
\end{matrix}
$$

$$
\begin{matrix}
 & & & & 1 & & & & \\
 & & & 1 & & 1 & & & \\
 & & 1 & & 2 & & 1 & & \\
 & 1 & & 3 & & 3 & & 1 & \\
1 & & 4 & & 6 & & 4 & & 1
\end{matrix}
$$



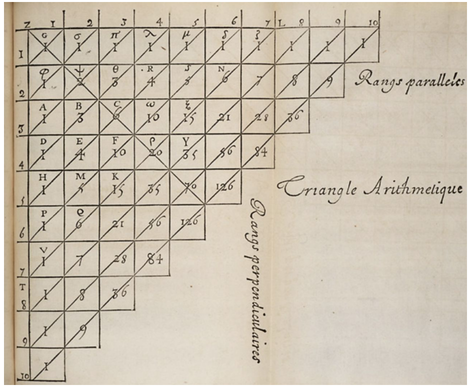
$$\binom{n}{0} = \binom{n}{n} = 1$$

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

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

Wikipedia: Pascal's Triangle.

Binomial Coefficients Properties



- Since:

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} \qquad \binom{n}{n-k} = \frac{n!}{(n-(n-k))!(n-k)!}$$

- We may optimize it:

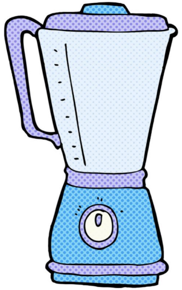
$$\binom{n}{k} = \begin{cases} \frac{n^{\underline{k}}}{k!} & \text{if } k \leq n/2 \\ \frac{n^{\underline{n-k}}}{(n-k)!} & \text{if } k > n/2 \end{cases}$$

```python
def comb(k,n):
    ans = 1
    if (k > n//2):
        k = n-k
    for i in range(k):
        ans *= (n-i) / (i+1)
    return ans
```


Video: How To Evaluate Binomial Coefficients.

Applying Combinations

- You have the ten fruits below (guess which one is which!), and you want to blend three of them to make a juice.
    - How many different juices can you make?
- Now you want to make a juice with three of the ten fruits, but your juice will be made with one, and only one, **red** fruit.
    - How many different juices can you make?

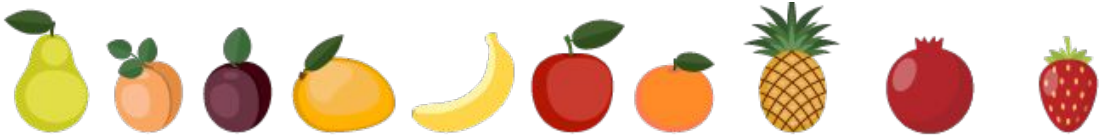


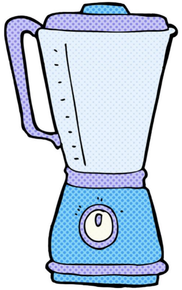
The answer is in the next slide, don't go there until you've tried to solve both problems.

Applying Combinations

- You have the ten fruits below (guess which one is which!), and you want to blend three of them to make a juice.
    - How many different juices can you make?

$$\binom{10}{3} = \frac{10 \times 9 \times 8}{3 \times 2 \times 1} = 120$$



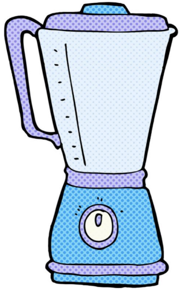


The next answer is in the next slide, don't go there until you've tried to solve the second problem.

Applying Combinations

- Now you want to make a juice with three of the ten fruits, but your juice will be made with one, and only one, **red** fruit.
  - How many different juices can you make?

$$\binom{3}{1} = 3 \qquad \times \qquad \binom{7}{2} = 21 \qquad = 63$$



Now imagine other problems that could be solved with computing combinations.

Combinations

Life is full of permutations and combinations. Sometimes the order you do things matters, sometimes it doesn't, but in order to find the solution in life you must work through each possibility presented to find your opportunity.
Gregory Willis

Given a set of elements, how many subsets of a given size can be generated?

And what would these subsets be?

As we will see, combinations can be computed considering elements which are unique - or not:

- k-combinations
- Binomial Coefficients
- **Multicombinations**

Multicombinations

$$\left(\!\!\binom{n}{k}\!\!\right)$$

Once again, the field uses several denominations to refer to the same thing, and sometimes give the same definitions to different things. We will use the term **Multicombinations** to describe the situation where you have a set of ***n*** elements and you want to express how many combinations of size ***k*** can be produced using as many number of each element as you want, for example:

- Given the set ***{A, B, C}***, how many subsets of two elements can be produced using as many number of each element as you want?
  - ***{A, A}***
  - ***{A, B}***
  - ***{A, C}***
  - ***{B, B}***
  - ***{B, C}***
  - ***{C, C}***

A.k.a.:

- ➔ Combinations with repetition
- ➔ Combination of multisets
- ➔ k-multicombinations

MERRIMACK COLLEGE

Wikipedia: Counting Multisets.

Multicombinations

- Given a set with ***n*** elements, the number of multicombinations of size ***k*** is given formally by the *multiset coefficient* which is similar to the binomial coefficient (only it uses double parenthesis, instead of single ones), which is defined numerically by a binomial coefficient over a larger set:

$$\left(\!\!\binom{n}{k}\!\!\right) = \binom{n+k-1}{k}$$

- For example, for ***n = 8*** and ***k = 3*** the multicombinations are given by:

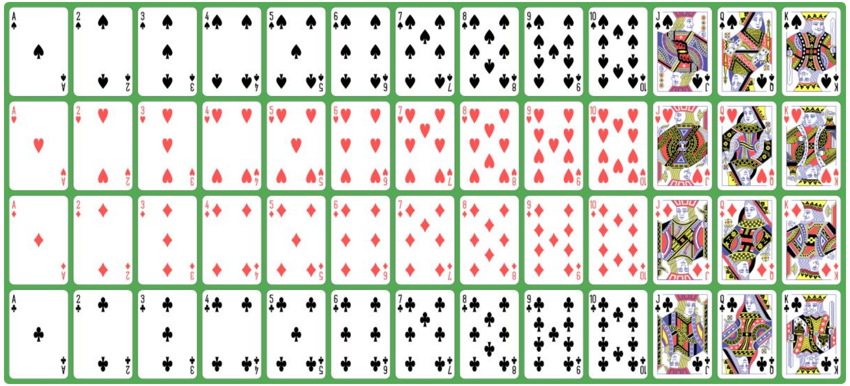
$$\left(\!\!\binom{8}{3}\!\!\right) = \binom{10}{3} = 120$$

Wikipedia: Counting Multisets.

Applying Multicombinations

- You have a deck with the usual 52 playing cards and 4 suits: 13 clubs, 13 diamonds, 13 hearts, and 13 spades.
  - If you draw three cards how many combinations of suits are possible (three clubs, two clubs and one diamond, etc...)?

- In this same deck, there are 4 aces, 4 twos, 4 threes, ... 4 tens, 4 jacks, 4 queens, and 4 kings.
  - If you draw three cards how many combinations of cards ignoring the suits are possible (three aces, two aces and one two, etc...)?



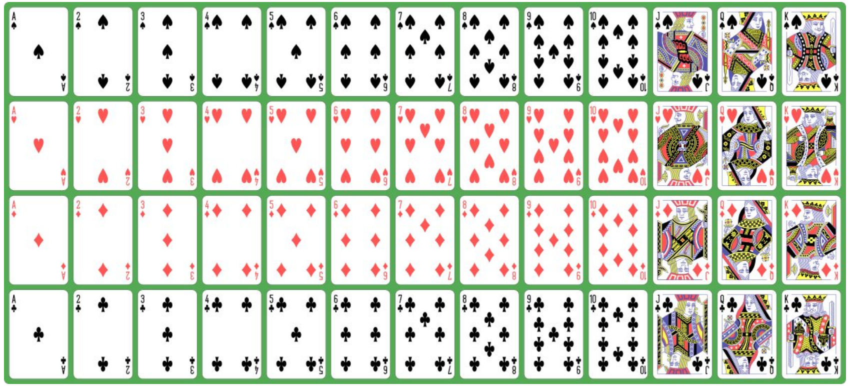
MERRIMACK COLLEGE

The answer is in the next slide, don't go there until you've tried to solve both problems.

Applying Multicombinations

- You have a deck with the usual 52 playing cards and 4 suits: 13 clubs, 13 diamonds, 13 hearts, and 13 spades.
    - If you draw three cards how many combinations of suits are possible (three clubs, two clubs and one diamond, etc...)?
        - Considering picking three cards out of four suits:

$$\left(\!\!\binom{4}{3}\!\!\right) = \binom{6}{3} = 20$$



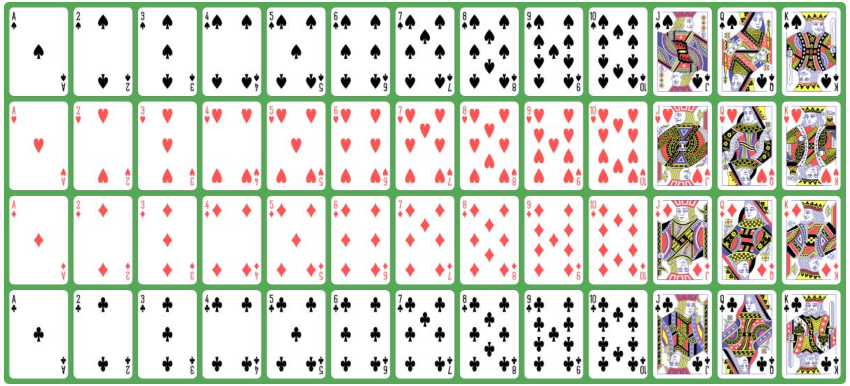
MERRIMACK COLLEGE

The next answer is in the next slide, don't go there until you've tried to solve the second problem.

Applying Multicombinations

- You have a deck with the usual 52 playing cards, 4 aces, 4 twos, 4 threes, … 4 tens, 4 jacks, 4 queens, and 4 kings.
  - If you draw three cards how many combinations of cards ignoring the suits are possible (three aces, two aces and one two, etc…)?
    - Considering picking three cards out of the 13 possible values:

$$\left(\!\!\binom{13}{3}\!\!\right) = \binom{15}{3} = 455$$



Now imagine other problems that could be solved with computing multicombinations.

Partitions

Gallia est omnis divisa in partes tres.

Julius Caesar

Given a set of elements, in how many ways can it be split into complementary partitions?

This is a kind of combinatorics problem that can be solved using what you know so far about combinations.

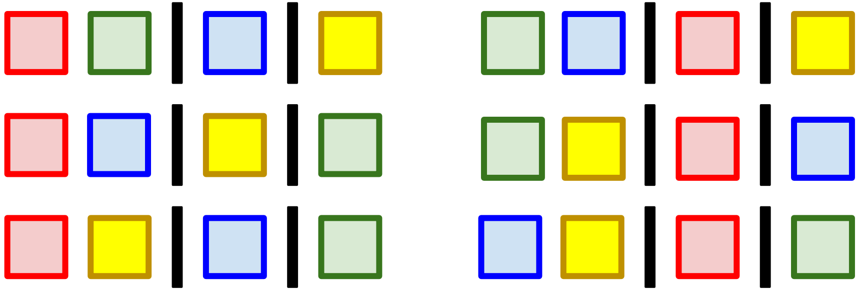
As we will see, partitions are complementary subsets, and since combinations are subsets, you can use them to find the possible partitions.

Partitions

Partitions are all about splitting a set of ***n*** elements into non-empty ***k*** subsets that are complementary.

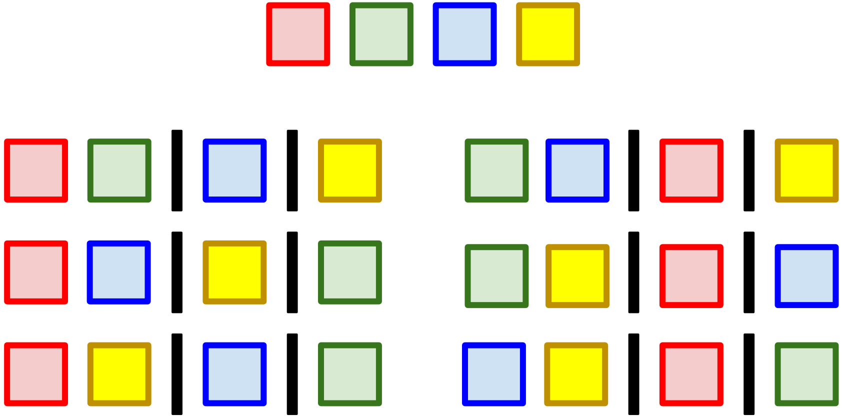
- Meaning, the ***k*** subsets have no intersection among them and the union is equal to the original set.
- For example, with ***n = 4*** and ***k = 3***,

the number of partition options is 6:



Wikipedia: Partition of a set.

Partitions



Partitions can be computed estimating the possible size of the subsets, for example, with ***n = 4*** and ***k = 3*** it will be necessary to define the three subsets with the following number of elements:

- One subset with 2 elements;
- One subset with one element;
- One subset with one element.

In this specific case, by choosing the two elements of the first set, necessarily the other two sets will be defined. Thus, the number of partition options will be given by combination of 2 out of 4:

$$\binom{4}{2} = 6$$

6 options

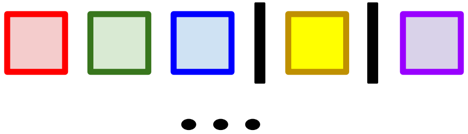
Wikipedia: Partition of a set.

Partitions

A more complex case would be the partition with ***n = 5*** and ***k = 3*** the three subsets with the following number of elements:

40 options

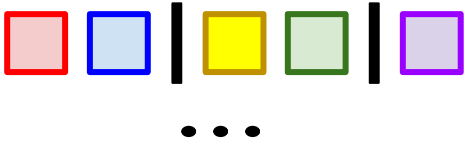
- One subset with 3 elements;
- One subset with one element;
- One subset with one element.



$$\binom{5}{3} = 10$$

Or:

- One subset with 2 elements;
- One subset with 2 elements;
- One subset with one element.



$$\binom{5}{2} \times \binom{3}{2} = 10 \times 3 = 30$$

Wikipedia: Partition of a set.

This Week's tasks

- Post discussion D#5
- Coding Project P#5
- Quiz Q#5

Tasks

- Post in the discussion one true and one false application of combinations.
- Coding Project #5, create a Pascal's Triangle.
- Quiz #5 about this week topics.

Post Discussion - Week 5 - D#5

Double work here!

Post in the discussion two practical problems:

- One that you could correctly solve using combinations;
- Another that cannot be correctly solved using combinations as seen.

However, do not say which one is which, as it will be the job of your colleagues to analyze the problems in order to say which one could be solved with combinations, and why. Additionally, you have to check out your colleagues' replies to see if they got it right or not.

Your task:

- Post your discussion in the message board by this Monday;
- Reply to your colleagues' posts in the message board by next Thursday.

This task counts towards the Discussions grade.

Fifth Coding Project - Week 5 - P#5

- Write a Python program which prints Pascal's Triangle.
  - Your program should ask the user for the number of rows (use a limit from 4 to 8 lines) of Pascal's Triangle to include.
  - Your program should have a function that computes the binomial coefficient of **n** and **k**, and this function should be used to find the values of the cells of the Pascal's Triangle.
  - The great challenge of this program is the output formatting which requires you to assemble the lines with the proper space.

Example in the next slide

Your task:

- Go to Canvas, and submit your Python file (.py) within the deadline:
  - The deadline for this assignment is next Thursday.

This assignment counts towards the Projects grade.

Fifth Coding Project - Week 5 - P#5

- For example:
  - If the user enters the value **4**, the output should be this:

```
      1
    1   1
  1   2   1
1   3   3   1
```

- If the user enters the value **6**, the output should be this:

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5  10  10   5   1
```

Your task:

- Go to Canvas, and submit your Python file (.py) within the deadline:
  - The deadline for this assignment is next Thursday.

This assignment counts towards the Projects grade.

Fifth Quiz - Week 5 - Q#5

- The fifth quiz in this course covers the topics of Week 5;
- The quiz will be available this Saturday, and it is composed of 10 questions;
- The quiz should be taken on Canvas (Module 5), and it is not timed:
  - You can take as long as you want to answer it;
- The quiz is open book, open notes, and you can even use any language interpreter to answer it;
- Yet, the quiz is evaluated and you are allowed to submit it only once.

Your task:

- Go to Canvas, answer the quiz and submit it within the deadline:
  - The deadline for the quiz is next Thursday.

This quiz counts towards the Quizzes grade.

We are in Week 5 of CSC 6000

- **Post discussion D#5 by Monday, reply to colleagues by Thursday;**
- **Do quiz Q#5 (available Saturday) by next Thursday;**
- **Develop coding project P#5 by next Thursday.**

Next Week - Linear Algebra: scalars, vectors, matrices, and tensors

MERRIMACK COLLEGE