Combinations

Alexander S. Kulikov

Steklov Mathematical Institute at St. Petersburg, Russian Academy of Sciences
and
University of California, San Diego

How Many?

- The main purpose is still to answer questions of the form “How many?”

How Many?

- The main purpose is still to answer questions of the form “How many?”
- We’ll consider several non-trivial problems and we’ll complement them with `Python` code

Outline

Previously on Combinatorics

Number of Games in a Tournament

Combinations

Rule of Sum

Rule of Sum

If there are n objects of the first type and there are k objects of the second type, then there are $n + k$ objects of one of two types

Rule of Sum

Rule of Sum

If there are n objects of the first type and there are k objects of the second type, then there are $n + k$ objects of one of two types

```
A = ['Alice', 'Bob', 'Charlie']
B = [0, 1, 2, 3]
print(A + B)
```

Rule of Sum

Rule of Sum

If there are n objects of the first type and there are k objects of the second type, then there are $n + k$ objects of one of two types

```
A = ['Alice', 'Bob', 'Charlie']
B = [0, 1, 2, 3]
print(A + B)
```

```
['Alice', 'Bob', 'Charlie', 0, 1, 2, 3]
```

Rule of Product

Rule of Product

If there are n objects of the first type and there are k objects of the second type, then there are $n \times k$ pairs of objects, the first of the first type and the second of the second type

Rule of Product

Rule of Product

If there are n objects of the first type and there are k objects of the second type, then there are $n \times k$ pairs of objects, the first of the first type and the second of the second type

```
from itertools import product
A = ['a', 'b']
B = [1, 2, 3]
print(list(product(A, B)))
```

Rule of Product

Rule of Product

If there are n objects of the first type and there are k objects of the second type, then there are $n \times k$ pairs of objects, the first of the first type and the second of the second type

```
from itertools import product
A = ['a', 'b']
B = [1, 2, 3]
print(list(product(A, B)))
```

```
[('a', 1), ('a', 2), ('a', 3), ('b', 1), ('b', 2), ('b', 3)]
```

Tuples

Tuples

The number of sequences of length k composed out of n symbols is n^k.

Tuples

Tuples

The number of sequences of length k composed out of n symbols is n^k.

```
from itertools import product
for p in product("ab", repeat=4):
    print("".join(p))
```

Tuples

Tuples

The number of sequences of length k composed out of n symbols is n^k.

```
from itertools import product
for p in product("ab", repeat=4):
    print("".join(p))
```

aaaa	abaa	baaa	bbaa
aaab	abab	baab	bbab
aaba	abba	baba	bbba
aabb	abbb	babb	bbbb

k-Permutations

k-Permutations

The number of sequences of length k with no repetitions composed out of n symbols is $n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}$.

k-Permutations

k-Permutations

The number of sequences of length k with no repetitions composed out of n symbols is $n(n-1)\cdots(n-k+1)=\frac{n!}{(n-k)!}$.

```
from itertools import permutations
for p in permutations("abcde", 2):
    print("".join(p))
```

k-Permutations

k-Permutations

The number of sequences of length k with no repetitions composed out of n symbols is $n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}$.

```
from itertools import permutations
for p in permutations("abcde", 2):
    print("".join(p))
```

ab	ba	ca	da	ea
ac	bc	cb	db	eb
ad	bd	cd	dc	ec
ae	be	ce	de	ed

Outline

Previously on Combinatorics

Number of Games in a Tournament

Combinations

Tournament

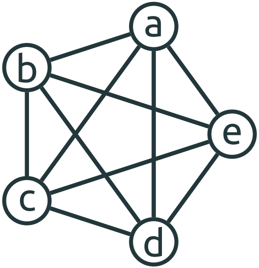
Number of games in a tournament

Five teams played a tournament: each team played with each other. What was the number of games?

Tournament

Number of games in a tournament

Five teams played a tournament: each team played with each other. What was the number of games?



Computing

- Each team played four games (with each of the other four teams)

Computing

- Each team played four games (with each of the other four teams)
- Hence, there were 5×4 games

Computing

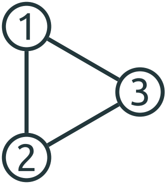
- Each team played four games (with each of the other four teams)
- Hence, there were 5×4 games
- Do you see a flaw in this argument?

Three Teams Instead of Five

- For three teams instead of five, our argument gives $3 \times 2 = 6$ games

Three Teams Instead of Five

- For three teams instead of five, our argument gives $3 \times 2 = 6$ games
- In fact, there are 3 games: during each game, one of the teams takes a rest



Closer Look

- Our argument says that each of five teams played with each of four other teams

Closer Look

- Our argument says that each of five teams played with each of four other teams
- Denote the five teams by a, b, c, d, and e and consider the games:

ab	ac	ad	ae
ba	bc	bd	be
ca	cb	cd	ce
da	db	dc	de
ea	eb	ec	ed

From a Different Point of View

- Let’s rearrange the games:

ab	ac	ad	ae	bc	bd	be	cd	ce	de
ba	ca	da	ea	cb	db	eb	dc	ec	ed

From a Different Point of View

- Let’s rearrange the games:

ab	ac	ad	ae	bc	bd	be	cd	ce	de
ba	ca	da	ea	cb	db	eb	dc	ec	ed

- Each game is counted twice!

From a Different Point of View

- Let’s rearrange the games:

ab	ac	ad	ae	bc	bd	be	cd	ce	de
ba	ca	da	ea	cb	db	eb	dc	ec	ed

- Each game is counted twice!
- Thus, the number of games is $(5 \times 4)/2 = 10$

Important Message

- When counting, make sure that each object is counted once

Important Message

- When counting, make sure that each object is counted once
- If *each* object is counted k times, divide the resulting count by k

Formally

Theorem

The number of games in a tournament with n teams (each pair of teams played each other exactly once) is $n(n-1)/2$.

Proof

- There are n choices of the first team in a game and $(n - 1)$ choices of the second team

Proof

- There are n choices of the first team in a game and $(n-1)$ choices of the second team
- Each game is counted twice: the game between teams i and j is counted as ij and as ji

Proof

- There are n choices of the first team in a game and $(n-1)$ choices of the second team
- Each game is counted twice: the game between teams i and j is counted as ij and as ji
- Thus, the total number of games is $n(n-1)/2$

Another Proof

- Let’s count the number of games recursively

Another Proof

- Let's count the number of games recursively
- Denote by $T(n)$ the number of games in a tournament with n teams

Another Proof

- Let's count the number of games recursively
- Denote by $T(n)$ the number of games in a tournament with n teams
- There are two types of games:

Another Proof

- Let's count the number of games recursively
- Denote by $T(n)$ the number of games in a tournament with n teams
- There are two types of games:
    1. games that involve the first team: $n - 1$

Another Proof

- Let's count the number of games recursively
- Denote by $T(n)$ the number of games in a tournament with n teams
- There are two types of games:
    1. games that involve the first team: $n-1$
    2. games that don't involve it: $T(n-1)$

Another Proof

- Let's count the number of games recursively
- Denote by $T(n)$ the number of games in a tournament with n teams
- There are two types of games:
    1. games that involve the first team: $n-1$
    2. games that don't involve it: $T(n-1)$
- Hence, $T(n) = (n-1) + T(n-1)$

Unwinding the Recurrence Relation

$$
\begin{aligned}
T(n) &= (n-1) + T(n-1) \\
&= (n-1) + (n-2) + T(n-2) \\
&= (n-1) + (n-2) + (n-3) + T(n-3) \\
&= \cdots \\
&= (n-1) + (n-2) + \cdots + 2 + 1 + 0
\end{aligned}
$$

Arithmetic Series

- Compute the sum of the series with itself reversed:

$$
\begin{array}{ccccc}
(n-1) & (n-2) & \cdots & 2 & 1 \\
1 & 2 & \cdots & (n-2) & (n-1) \\
\hline
n & n & n & n & n
\end{array}
$$

Arithmetic Series

- Compute the sum of the series with itself reversed:

$$
\begin{array}{ccccc}
(n-1) & (n-2) & \cdots & 2 & 1 \\
1 & 2 & \cdots & (n-2) & (n-1) \\
\hline
n & n & n & n & n
\end{array}
$$

- Hence, $T(n) = n(n-1)/2$

Code

```
from itertools import combinations

for c in combinations("abcdefgh", 2):
    print("".join(c))
```

Code

```
from itertools import combinations

for c in combinations("abcdefgh", 2):
    print("".join(c))
```

```
ab
ac
ad
ae
af
ag
ah
```

```
bc
bd
be
bf
bg
bh
cd
```

```
ce
cf
cg
ch
de
df
dg
```

```
dh
ef
eg
eh
fg
fh
gh
```

Recursion

```
def T(n):
    if n <= 1:
        return 0

    return (n - 1) + T(n - 1)

print(T(8))
```

Recursion

```python
def T(n):
    if n <= 1:
        return 0

    return (n - 1) + T(n - 1)

print(T(8))
```

```
28
```

Outline

Previously on Combinatorics

Number of Games in a Tournament

Combinations

Counting Subsets

Journey

You are organizing a car journey. You have five friends, but there are only three vacant places in you car. What is the number of ways of taking three of your five friends to the journey?

Counting Subsets

Journey

You are organizing a car journey. You have five friends, but there are only three vacant places in you car. What is the number of ways of taking three of your five friends to the journey?

Subsets

What is the number of ways of choosing 3 elements out of a set of size 5?

Counting

- There are five choices of the first friend, four choices of the second friend, and three choices of the third friend

Counting

- There are five choices of the first friend, four choices of the second friend, and three choices of the third friend
- In total $5 \times 4 \times 3 = 60$ choices

Counting

- There are five choices of the first friend, four choices of the second friend, and three choices of the third friend
- In total $5 \times 4 \times 3 = 60$ choices
- But each group of three friends is counted $3! = 6$ times: a group {a, b, c} is counted as abc, acb, bac, bca, cab, cba

Counting

- There are five choices of the first friend, four choices of the second friend, and three choices of the third friend
- In total $5 \times 4 \times 3 = 60$ choices
- But each group of three friends is counted $3! = 6$ times: a group {a, b, c} is counted as abc, acb, bac, bca, cab, cba
- Thus, the answer is $(5 \times 4 \times 3)/3! = 10$

Code

```
import itertools as it

for c in it.combinations("abcde", 3):
    print("".join(c))
```

Code

```
import itertools as it

for c in it.combinations("abcde", 3):
    print("".join(c))
```

```
abc
abd
abe
acd
ace
```

```
ade
bcd
bce
bde
cde
```

Combinations

Definition

For a set S, its k-combination is a subset of S of size k.

Combinations

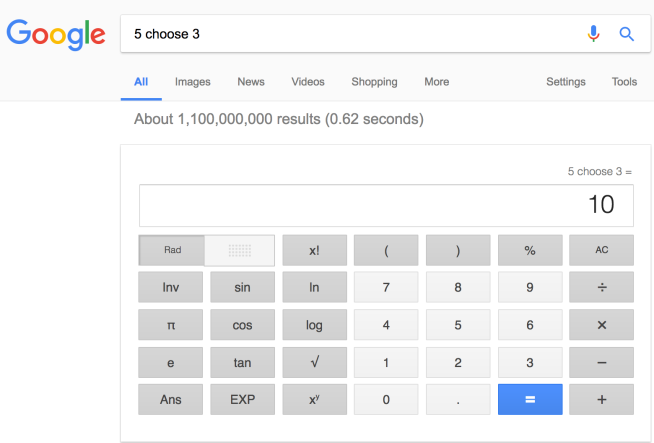
Definition

For a set S, its k-combination is a subset of S of size k.

Definition

The number of k-combinations of an n element set is denoted by $\binom{n}{k}$. Pronounced: "n choose k".

$\binom{5}{3}$ by Google



3-Combinations and 3-Permutations

abc	abd	abe	acd	ace	ade	bcd	bce	bde	cde
acb	adb	aeb	adc	aec	aed	bdc	bec	bed	ced
bac	bad	bae	cad	cae	dae	cbd	cbe	dbe	dce
bca	bda	bea	cda	cea	dea	cdb	ceb	deb	dec
cba	dba	eba	dca	eca	eda	dcb	ecb	edb	edc
cab	dab	eab	dac	eac	ead	dbc	ebc	ebd	ecd

3-Combinations and 3-Permutations

	abc	abd	abe	acd	ace	ade	bcd	bce	bde	cde
	acb	adb	aeb	adc	aec	aed	bdc	bec	bed	ced
	bac	bad	bae	cad	cae	dae	cbd	cbe	dbe	dce
3!	bca	bda	bea	cda	cea	dea	cdb	ceb	deb	dec
	cba	dba	eba	dca	eca	eda	dcb	ecb	edb	edc
	cab	dab	eab	dac	eac	ead	dbc	ebc	ebd	ecd

3-Combinations and 3-Permutations

$\binom{5}{3}$

$3!$

abc	abd	abe	acd	ace	ade	bcd	bce	bde	cde
acb	adb	aeb	adc	aec	aed	bdc	bec	bed	ced
bac	bad	bae	cad	cae	dae	cbd	cbe	dbe	dce
bca	bda	bea	cda	cea	dea	cdb	ceb	deb	dec
cba	dba	eba	dca	eca	eda	dcb	ecb	edb	edc
cab	dab	eab	dac	eac	ead	dbc	ebc	ebd	ecd

3-Combinations and 3-Permutations

$\binom{5}{3}$

$3!$

abc	abd	abe	acd	ace	ade	bcd	bce	bde	cde
acb	adb	aeb	adc	aec	aed	bdc	bec	bed	ced
bac	bad	bae	cad	cae	dae	cbd	cbe	dbe	dce
bca	bda	bea	cda	cea	dea	cdb	ceb	deb	dec
cba	dba	eba	dca	eca	eda	dcb	ecb	edb	edc
cab	dab	eab	dac	eac	ead	dbc	ebc	ebd	ecd

$$3!\binom{5}{3} = \frac{5!}{(5-3)!}$$

Number of Combinations

Theorem

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

Proof

- Need to count the number of subsets of size k of a set of size n

Proof

- Need to count the number of subsets of size k of a set of size n
- There are n choices of the first element, $(n-1)$ choices of the second element, $\ldots$, $(n-k+1)$ choices of the k-th element

Proof

- Need to count the number of subsets of size k of a set of size n
- There are n choices of the first element, $(n-1)$ choices of the second element, $\ldots$, $(n-k+1)$ choices of the k-th element
- This gives $n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}$

Proof

- Need to count the number of subsets of size k of a set of size n
- There are n choices of the first element, $(n-1)$ choices of the second element, $\ldots$, $(n-k+1)$ choices of the k-th element
- This gives $n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}$
- But this is the number of k-permutations rather than the number of k-combinations

Proof

- Need to count the number of subsets of size k of a set of size n
- There are n choices of the first element, $(n-1)$ choices of the second element, $\ldots$, $(n-k+1)$ choices of the k-th element
- This gives $n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}$
- But this is the number of k-permutations rather than the number of k-combinations
- Each subset is counted $k!$ times

Proof

- Need to count the number of subsets of size k of a set of size n
- There are n choices of the first element, $(n-1)$ choices of the second element, $\dots$, $(n-k+1)$ choices of the k-th element
- This gives $n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}$
- But this is the number of k-permutations rather than the number of k-combinations
- Each subset is counted $k!$ times
- This finally gives $\frac{n!}{k!(n-k)!}$