CM1103 PROBLEM SOLVING WITH PYTHON

COUNTING

Dr Jing Wu
School of Computer Science & Informatics
Cardiff University

Example

How many ways are there that two prize winners can be chosen from the set { Andrew, Bob, Clare, Deb}?

Example

How many outcomes are possible from rolling two dice?
Does it matter what colour they are?

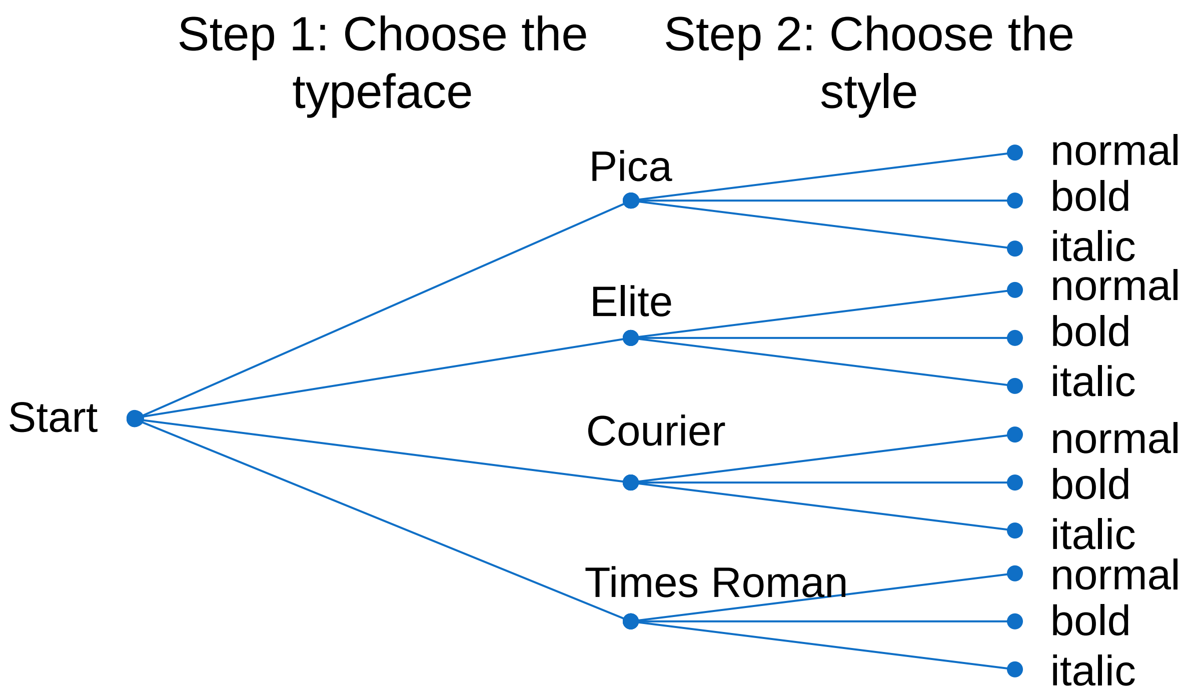
Overview

- Rules for counting
  - The product rule
  - The sum rule
- What are we counting
  - Permutations
  - Combinations

Example

Suppose that a printer can print in four typefaces: Pica, Elite, Courier, and Times Roman. Suppose also that each typeface can appear in one of three styles: normal, bold, and italic. A font is a combination of a particular typeface with a particular style, such as Courier Italic. How many fonts are there?

Imagine the pairing of the typeface and style as a two-step operation:



Altogether: $4 \times 3 = 12$ fonts

The product rule

If an operation consists of k steps, where:

- 1st step can be performed in n_1 ways;
- 2nd step can be performed in n_2 ways
  regardless of how 1st step was performed;
- ...
- kth step can be performed in n_k ways
  regardless of how preceding steps were performed;

Then the entire operation can be performed in $n_1 \times n_2 \times \cdots \times n_k$ ways.

Example

Example: the number of elements in a Cartesian product

Suppose A_1, A_2, A_3 are sets with n_1, n_2, n_3 elements, respectively. How many elements are there in the set $A_1 \times A_2 \times A_3$?

$A_1 \times A_2 \times A_3 = \{(a_1, a_2, a_3) \colon a_1 \in A_1, a_2 \in A_2, and\ a_3 \in A_3\}$

Imaging the construction of the tuple (a_1, a_2, a_3) as a three step operation:

Step 1: choose a_1 from A_1; There are n_1 ways.

Step 2: choose a_2 from A_2; There are n_2 ways.

Step 3: choose a_3 from A_3. There are n_3 ways.

By the product rule, there are $n_1 \times n_2 \times n_3$ ways to construct the tuple.

Therefore, there are $n_1 \times n_2 \times n_3$ elements in $A_1 \times A_2 \times A_3$.

Extension to $A_1, A_2, \ldots, A_n$:

$|A_1 \times A_2 \times \cdots \times A_n| = |A_1| \times |A_2| \times \cdots \times |A_n|$

Example

Example: two digit numbers

How many integer numbers (in decimal) can be represented using two digits?

$$a_1 a_2$$

Apply product rule:

Step 1: choose a digit from $\{0, \cdots, 9\}$ for a_1, there are 10 choices;

Step 2: choose a digit from $\{0, \cdots, 9\}$ for a_2, there are 10 choices, regardless of the choice for a_1;

In total, there are $10 \times 10 = 100$ ways to choose the two digits.

So, the two digits can represent 100 numbers, from 00 to 99.

? For the same problem, if a_1, a_2 subject to a constraint: $a_1 + a_2 \leq 15$, can we still apply product rule?

No, because the number of choices for a_2 will depend on the choice for a_1.

Example

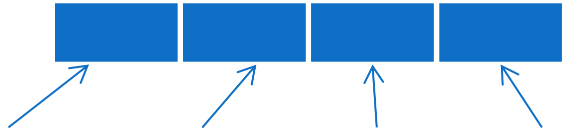
Example: bank pins

If a bank pin consists of 4 digits (a digit is a number from 0 to 9), how many possible pins are there?

$$10 \times 10 \times 10 \times 10 = 10000$$

Example: bank pins

If a bank pin consists of 4 different digits, how many possible pins are there?



$$10 \times 9 \times 8 \times 7 = 5040$$

Example

Example: nested loops

```
------------------------------------------------------------------------------------
for i in range(0,4):
    for j in range(0,3):
        # Do some work
------------------------------------------------------------------------------------
```

How many times will *some work* be executed?

range(0,4): [0, 1, 2, 3]

range(0,3): [0, 1, 2]

The total number of *some work* being executed: $4 \times 3 = 12$.

i	0			1			2			3		
j	0	1	2	0	1	2	0	1	2	0	1	2

3 + 3 + 3 + 3 = 12

The sum rule

If $\{S_1, S_2, \cdots, S_k\}$ is a *partition* (see Sets handout) of a finite set S, then:

$$|S| = |S_1| + |S_2| + \cdots + |S_k|$$

Example

Example: password

Suppose we insist users choose a password of exactly 4 characters with a pattern: numbers followed by letters. There should be at least one number and at least one letter (only numbers & letters allowed). How many possible passwords are there?

Let S be the set of allowed passwords,

then the following three sets form a partition of S:

$S_1 =$ {passwords with 1 number followed by 3 letters}

$S_2 =$ {passwords with 2 numbers followed by 2 letters}

$S_3 =$ {passwords with 3 numbers followed by 1 letter}

By rule of sums:

product rule

$$|S| = |S_1| + |S_2| + |S_3|$$

$$= 10 \times 26 \times 26 \times 26 + 10 \times 10 \times 26 \times 26 + 10 \times 10 \times 10 \times 26$$

What if there's no restriction on order?

Example

Example: two digit numbers

How many integer numbers (in decimal) can be represented using two digits?
$a_1 a_2$, if a_1, a_2 subject to a constraint: $a_1 + a_2 \leq 15$.

Let S be the set of possible integers. A partition of S:

$$S_1 = \{a_1 a_2 \colon a_1 = 0, 0 \leq a_2 \leq 9\}$$
$$S_2 = \{a_1 a_2 \colon a_1 = 1, 0 \leq a_2 \leq 9\}$$
$$S_3 = \{a_1 a_2 \colon a_1 = 2, 0 \leq a_2 \leq 9\}$$
$$S_4 = \{a_1 a_2 \colon a_1 = 3, 0 \leq a_2 \leq 9\}$$
$$S_5 = \{a_1 a_2 \colon a_1 = 4, 0 \leq a_2 \leq 9\}$$
$$S_6 = \{a_1 a_2 \colon a_1 = 5, 0 \leq a_2 \leq 9\}$$
$$S_7 = \{a_1 a_2 \colon a_1 = 6, 0 \leq a_2 \leq 9\}$$
$$S_8 = \{a_1 a_2 \colon a_1 = 7, 0 \leq a_2 \leq 8\}$$
$$S_9 = \{a_1 a_2 \colon a_1 = 8, 0 \leq a_2 \leq 7\}$$
$$S_{10} = \{a_1 a_2 \colon a_1 = 9, 0 \leq a_2 \leq 6\}$$

So, $|S| = |S_1| + |S_2| + |S_3| + |S_4| + |S_5| + |S_6| + |S_7| + |S_8| + |S_9| + |S_{10}|$

$$= 10 + 10 + 10 + 10 + 10 + 10 + 10 + 9 + 8 + 7 = 94$$

Permutations

Definition (permutation)

A permutation of a set of objects is an order of the objects in sequence.

Example:

$S = \{a, b, c\}$

Permutations of S:

$a\ b\ c \qquad a\ c\ b \qquad b\ a\ c \qquad b\ c\ a \qquad c\ a\ b \qquad c\ b\ a$

Counting permutations

A set A of n objects has $n! = n \times (n-1) \times (n-2) \times \cdots \times 1$ permutations.

Why?
We have:
n choices from A for the 1st element p_1 of the permutation
$n-1$ choices from $A - \{p_1\}$ for the 2nd element p_2 of the permutation
$n-2$ choices from $A - \{p_1, p_2\}$ for the 3rd element p_3 of the permutation
$\dots\dots$
1 choice for the last
so follows by product rule

Example

Example: permutations of the letters in a word

How many ways can the letters in the word *COMPUTER* be arranged in a row?

The number of permutations of the set $S = \{'C', 'O', 'M', 'P', 'U', 'T', 'E', 'R'\}$

$8! = 40320$

Example: permutations of the letters in a word

How many ways can the letters in the word *COMPUTER* be arranged if the letters *CO* must remain next to each other (in order) as a unit?

The number of permutations of the set $S = \{'CO', 'M', 'P', 'U', 'T', 'E', 'R'\}$

$7! = 5040$

r-permutations

Definition (r-permutation)

An r-permutation of a set of n elements is an ordered selection of r elements taken from the set without repetition.

Definition ($P(n, r)$)

The number of r-permutations of a set of n elements is given by:

$$P(n,r) = \frac{n!}{(n-r)!}$$

Example

Suppose we have a squad of 14 cricketeers. How many batting line-ups of 11 players are there?

$$P(14,11) = \frac{14!}{(14-11)!} = \frac{14!}{3!}$$

r-permutations

Why $P(n, r) = \frac{n!}{(n-r)!}$?

To calculate $P(n, r)$ for a set A we have:

n choices from A for the 1st element p_1 of the r-permutation

$n - 1$ choices from $A - \{p_1\}$ for the 2nd element p_2 of the r-permutation

$n - 2$ choices from $A - \{p_1, p_2\}$ for the 3rd element p_3 of the r-permutation

$\dots\dots$

$n - (r - 1)$ choices for the r^{th} element of the r-permutation

So, follows by product rule:

$$
\begin{aligned}
P(n, r) &= n \times (n-1) \times (n-2) \times \cdots \times \big(n - (r-1)\big) \\
&= \frac{n \times (n-1) \times (n-2) \times \cdots \times \big(n - (r-1)\big) \times (n-r) \times \cdots \times 1}{(n-r) \times \cdots \times 1} \\
&= \frac{n!}{(n-r)!}
\end{aligned}
$$

Permutations in Python

```
----------------------------------------------------------------
>>> import itertools
>>> for perm in itertools.permutations("abcd"):
...         print(perm)
----------------------------------------------------------------
----------------------------------------------------------------
>>> len(list(itertools.permutations("123")))
6
>>> list(itertools.permutations("123"))
[('1', '2', '3'), ('1', '3', '2'), ('2', '1', '3'), ('2', '3', '1'), ('3', '1', '2'), ('3', '2', '1')]
----------------------------------------------------------------
----------------------------------------------------------------
>>> for perm in itertools.permutations("abcd", 2):
...         print(perm)
----------------------------------------------------------------
```

r-combinations

Definition (r-combination)

An r-combination of a set of size n is a subset of size r

Definition ($C(n,r)$)

Let $C(n,r)$ denote the number of r-combinations of a set of size n. Then:

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

Why?

An r-permutation consists of two steps:

selection of r elements to form a subset of size r	$C(n,r)$
order the r elements into a sequence	$P(r,r)$

Apply product rule: $C(n,r) \times P(r,r) = P(n,r)$

So $C(n,r) = \frac{P(n,r)}{P(r,r)} = \frac{n!}{r!(n-r)!}$

Example

How many different ways are there of choosing a team of 11 football players from a squad of 23?

Compared to the *line-up* of cricket players problem, here we don't care about positions of the players.

$$C(23,11) = \frac{23!}{11! \times (23-11)!} = \frac{23!}{11! \times 12!}$$

What if we require a goalkeeper to be chosen, and there are 3 in the squad?

There are two step:

Choose the goal keeper $C(3,1)$

Choose the other players $C(20,10)$

Follow product rule:

$$C(3,1) \times C(20,10) = \frac{3!}{1! \times 2!} \times \frac{20!}{10! \times 10!} = \frac{3 \times 20!}{10! \times 10!}$$

Example

Example: password

Suppose we insist users choose a password of exactly 4 characters. There should be at least one number and at least one letter (only numbers & letters allowed). How many possible passwords are there?

The password example, but with no restriction on order of numbers/letters.

Let S be the set of allowed passwords. A partition of S:

$$S_1 = \{\text{passwords with 1 number}\}$$

$$S_2 = \{\text{passwords with 2 numbers}\}$$

$$S_3 = \{\text{passwords with 3 numbers}\}$$

By rule of sums:

$$|S| = |S_1| + |S_2| + |S_3|$$

$$= C(4,1) \times 10 \times 26 \times 26 \times 26$$

$$+C(4,2) \times 10 \times 10 \times 26 \times 26$$

$$+C(4,3) \times 10 \times 10 \times 10 \times 26$$

Combinatorial equivalence

Some counting problems can be made easier by recognising when there is a one-to-one correspondence to another problem

Example	
How many ways are there of choosing 2 elements from the set $\{1, 2, \cdots, 9\}$?	$C(9,2) = \frac{9!}{2!\,(9-2)!} = \frac{9!}{2!\,7!}$
How many ways are there of choosing 7 elements from the set $\{1, 2, \cdots, 9\}$?	$C(9,7) = \frac{9!}{7!\,(9-7)!} = \frac{9!}{7!\,2!}$

$C(n,r) = C(n,(n-r))$

r-combinations in Python

```
-----------------------------------------------------------------------------------------------------
>>> for comb in itertools.combinations(range(1,5),3):
...         print(comb)
-----------------------------------------------------------------------------------------------------
```

Repetition

Theorem (Permutations with repetition)

The number of r-permutations of a set of size n is n^r if repetition is allowed.

Apply product rule

Theorem (Combinations with repetition)

The number of r-combinations of a set of size n is $C(n+r-1, r)$ if repetition is allowed.

Example:

$S = \{a, b, c\}$

2-combinations of S (repetition allowed):

Why $C(n+r-1,r)$?

a		b		c	
	|		|		
x	|	x	|		$\{a,b\}$
x	|		|	x	$\{a,c\}$
	|	x	|	x	$\{b,c\}$
x x	|		|		$\{a,a\}$
	|	x x	|		$\{b,b\}$
	|		|	x x	$\{c,c\}$

Draw lines between objects
$n-1$ lines

Put a cross below the objects which are selected

r crosses

In total, there are $n-1+r$ lines and crosses

We regard them as $n+r-1$ positions

The selection with repetition problem can be translated as:
Among the $n+r-1$ positions, choose r for crosses.

$$C(n+r-1,r)$$

Example

How many different strings of length 5 can be formed using only consonants?

A doughnut shop sells 4 different types of doughnut. How many different ways are there of choosing 6 doughnuts?

Indistinguishable objects

Theorem (Permutations with indistinguishable objects)

Suppose we have n_1 objects of *type* 1, n_2 of *type* 2, $\dots$, n_k of type k with $n = n_1 + \cdots + n_k$. The number of permutations of these objects is:

$$\frac{n!}{n_1!\, n_2! \cdots n_k!}$$

Example

How many strings can be made by rearranging the letters of **scarceness**?

Why $\frac{n!}{n_1!n_2!\cdots n_k!}$?

n_1 objects of *type* 1, n_2 objects of *type* 2, …, n_k objects of type k

$n = n_1 + n_2 + \cdots + n_k$

n positions

Step 1: from the n positions, select n_1 for *type* 1 $\qquad C(n, n_1)$

Step 2: from the remaining $n - n_1$ positions, select n_2 for *type* 2

… $\qquad C(n - n_1, n_2)$

Step k: from the remaining $n - n_1 - \cdots - n_{k-1} = n_k$ positions, select n_k for *type* k $\qquad C(n - n_1 - \cdots - n_{k-1}, n_k)$

Follow product rule:

$$C(n, n_1) \times C(n - n_1, n_2) \times \cdots \times C(n - n_1 - \cdots - n_{k-1}, n_k)$$

$$= \frac{n!}{n_1!\cancel{(n-n_1)!}} \times \frac{\cancel{(n-n_1)!}}{n_2!\cancel{(n-n_1-n_2)!}} \times \frac{\cancel{(n-n_1-n_2)!}}{n_3!\cancel{(n-n_1-n_2-n_3)!}} \times \cdots \times \frac{\cancel{(n-n_1-\cdots-n_{k-1})!}}{n_k!0!}$$

$$= \frac{n!}{n_1!n_2!\cdots n_k!}$$

Summary

You should be able to:

- Count and list the permutations, r-permutations, and r-combinations of sets by hand, by formula and using Python
- Count permutations and combinations when repetition is allowed or there are indistinguishable objects in the set
- Be able to solve counting problem by using the product and sum rules to divide and combine answers

Definitions covered

- Permutation, r-permutation, r-combination
- $P(n, r)$, $C(n, r)$