Introduction to Course Content

Overview of the Course

Week 1	Propositional Logic
Week 2	Predicate Logic
Week 3	Proofs
Week 4	Sets, Functions, Relations
Week 5	Relations, Sequences, Summations
Week 6	Algorithms
Week 7	Complexity of Algorithms
Week 8	Number Theory
Week 9	Induction and Recursion
Week 10	Counting
Week 11	Advanced Counting
Week 12	Probability
Week 13	Advanced Probability

Example 1: Swiss Cities

- We find a lot of tabular data, e.g. https://www.citypopulation.de/de/switzerland/cities/

Name	Abk.	Hauptstadt	Fläche A (km²)	Einwohner Zensus (C) 1980-12-02	Einwohner Zensus (C) 1990-12-04	Einwohner Zensus (C) 2000-12-05	Einwohner Schätzung (E) 2010-12-31	Einwohner Schätzung (E) 2019-12-31
Graubünden	GR	Chur	7.105	164.641	173.890	187.058	192.621	199.021
Bern	BE	Bern	5.941	898.397	943.196	957.197	979.802	1.039.474
Valais [Wallis]	VS	Sion	5.225	218.707	249.817	272.399	312.684	345.525
Vaud [Waadt]	VD	Lausanne	3.212	528.747	601.816	640.657	713.281	805.098
Ticino [Tessin]	TI	Bellinzona	2.812	265.899	282.181	306.846	333.753	351.491
St. Gallen	SG	St. Gallen	2.026	391.995	427.501	452.837	478.907	510.734
Zürich	ZH	Zürich	1.729	1.122.839	1.179.044	1.247.906	1.373.068	1.539.275
Fribourg [Freiburg]	FR	Fribourg	1.671	185.246	213.571	241.706	278.493	321.783
Luzern	LU	Luzern	1.493	296.159	326.268	350.504	377.610	413.120
Aargau	AG	Aarau	1.404	453.442	507.508	547.493	611.466	685.845
Uri	UR	Altdorf	1.077	33.883	34.208	34.777	35.422	36.703
Thurgau	TG	Frauenfeld	991	183.795	209.362	228.875	248.444	279.547
Schwyz	SZ	Schwyz	908	97.354	111.964	128.704	146.730	160.480
Jura	JU	Delémont	836	64.986	66.163	68.224	70.032	73.584
Neuchâtel [Neuenburg]	NE	Neuchâtel	803	158.368	163.985	167.949	172.085	176.496
Solothurn	SO	Solothurn	791	218.102	231.746	244.341	255.284	275.247
Glarus	GL	Glarus	685	36.718	38.508	38.183	38.608	40.590
Basel-Landschaft	BL	Liestal	518	233.447	248.484	259.374	274.404	289.468
Obwalden	OW	Sarnen	490	25.865	29.025	32.427	35.585	37.930
Schaffhausen	SH	Schaffhausen	298	69.413	72.160	73.392	76.356	82.348
Genève [Genf]	GE	Genève	282	349.040	379.190	413.673	457.715	504.128
Nidwalden	NW	Stans	276	28.617	33.044	37.235	41.024	43.087
Appenzell Ausserrhoden	AR	Herisau	243	47.611	52.229	53.504	53.017	55.445
Zug	ZG	Zug	239	75.930	85.546	100.052	113.105	127.642
Appenzell Innerrhoden	AI	Appenzell	173	12.844	13.870	14.618	15.688	16.128
Basel-Stadt	BS	Basel	37	203.915	199.411	188.079	184.950	195.844
Schweiz	**CHE**	**Bern**	**41.285**	**6.365.960**	**6.873.687**	**7.288.010**	**7.870.134**	**8.603.899**

Load in Excel

ordered tuple

relation (a **set**)

Canton	Acronym	Capital	Surface	Population
Appenzell Innerrhoden	AI	Appenzell	173	16128
Uri	UR	Altdorf	1077	36703
Obwalden	OW	Sarnen	490	37930
Glarus	GL	Glarus	685	40590
Nidwalden	NW	Stans	276	43087
Appenzell Ausserrhoden	AR	Herisau	243	55445
Jura	JU	Delémont	836	73584
Schaffhausen	SH	Schaffhausen	298	82348
Zug	ZG	Zug	239	127642
Schwyz	SZ	Schwyz	908	160480
Neuchâtel [Neuenburg]	NE	Neuchâtel	803	176496
Basel-Stadt	BS	Basel	37	195844
Graubünden	GR	Chur	7105	199021
Solothurn	SO	Solothurn	791	275247
Thurgau	TG	Frauenfeld	991	279547
Basel-Landschaft	BL	Liestal	518	289468
Fribourg [Freiburg]	FR	Fribourg	1671	321783
Valais [Wallis]	VS	Sion	5225	345525
Ticino [Tessin]	TI	Bellinzona	2812	351491
Luzern	LU	Luzern	1493	413120
Genève [Genf]	GE	Genève	282	504128
St Gallen	SG	St Gallen	2026	510734
Aargau	AG	Aarau	1404	685845
Vaud [Waadt]	VD	Lausanne	3212	805098
Bern	BE	Bern	5941	1039474
Zürich	ZH	Zürich	1729	1539275

Week 1	Propositional Logic
Week 2	Predicate Logic
Week 4	Sets, Functions, Relations

Load in Excel

predicate **propositional connective**

ordered tuple

relation (a **set**)

truth table

Canton	Acronym	Capital	Surface	Population		Surface > 1000	Population < 100000	Surface > 1000 AND Population < 1000000
Appenzell Innerrhoden	AI	Appenzell	173	16128	0	FALSE	TRUE	FALSE
Uri	UR	Altdorf	1077	36703	1	TRUE	TRUE	TRUE
Obwalden	OW	Sarnen	490	37930	0	FALSE	TRUE	FALSE
Glarus	GL	Glarus	685	40590	0	FALSE	TRUE	FALSE
Nidwalden	NW	Stans	276	43087	0	FALSE	TRUE	FALSE
Appenzell Ausserrhoden	AR	Herisau	243	55445	0	FALSE	TRUE	FALSE
Jura	JU	Delémont	836	73584	0	FALSE	TRUE	FALSE
Schaffhausen	SH	Schaffhausen	298	82348	0	FALSE	TRUE	FALSE
Zug	ZG	Zug	239	127642	0	FALSE	FALSE	FALSE
Schwyz	SZ	Schwyz	908	160480	0	FALSE	FALSE	FALSE
Neuchâtel [Neuenburg]	NE	Neuchâtel	803	176496	0	FALSE	FALSE	FALSE
Basel-Stadt	BS	Basel	37	195844	0	FALSE	FALSE	FALSE
Graubünden	GR	Chur	7105	199021	0	TRUE	FALSE	FALSE
Solothurn	SO	Solothurn	791	275247	0	FALSE	FALSE	FALSE
Thurgau	TG	Frauenfeld	991	279547	0	FALSE	FALSE	FALSE
Basel-Landschaft	BL	Liestal	518	289468	0	FALSE	FALSE	FALSE
Fribourg [Freiburg]	FR	Fribourg	1671	321783	0	TRUE	FALSE	FALSE
Valais [Wallis]	VS	Sion	5225	345525	0	TRUE	FALSE	FALSE
Ticino [Tessin]	TI	Bellinzona	2812	351491	0	TRUE	FALSE	FALSE
Luzern	LU	Luzern	1493	413120	0	TRUE	FALSE	FALSE
Genève [Genf]	GE	Genève	282	504128	0	FALSE	FALSE	FALSE
St Gallen	SG	St Gallen	2026	510734	0	TRUE	FALSE	FALSE
Aargau	AG	Aarau	1404	685845	0	TRUE	FALSE	FALSE
Vaud [Waadt]	VD	Lausanne	3212	805098	0	TRUE	FALSE	FALSE
Bern	BE	Bern	5941	1039474	0	TRUE	FALSE	FALSE
Zürich	ZH	Zürich	1729	1539275	0	TRUE	FALSE	FALSE

Week 1	Propositional Logic
Week 2	Predicate Logic
Week 4	Sets, Functions, Relations

Functions

Canton	Acronym	Capital	Surface	Population		Surface > 1000	Population < 100000	Surface > 1000 AND Population < 1000000
Appenzell Innerrhoden	AI	Appenzell	173	16128	0	FALSE	TRUE	FALSE
Uri	UR	Altdorf	1077	36703	1	TRUE	TRUE	TRUE
Obwalden	OW	Sarnen	490	37930	0	FALSE	TRUE	FALSE
Glarus	GL	Glarus	685	40590	0	FALSE	TRUE	FALSE
Nidwalden	NW	Stans	276	43087	0	FALSE	TRUE	FALSE
Appenzell Ausserrhoden	AR	Herisau	243	55445	0	FALSE	TRUE	FALSE
Jura	JU	Delémont	836	73584	0	FALSE	TRUE	FALSE
Schaffhausen	SH	Schaffhausen	298	82348	0	FALSE	TRUE	FALSE
Zug	ZG	Zug	239	127642	0	FALSE	FALSE	FALSE
Schwyz	SZ	Schwyz	908	160480	0	FALSE	FALSE	FALSE
Neuchâtel [Neuenburg]	NE	Neuchâtel	803	176496	0	FALSE	FALSE	FALSE
Basel-Stadt	BS	Basel	37	195844	0	FALSE	FALSE	FALSE
Graubünden	GR	Chur	7105	199021	0	TRUE	FALSE	FALSE
Solothurn	SO	Solothurn	791	275247	0	FALSE	FALSE	FALSE
Thurgau	TG	Frauenfeld	991	279547	0	FALSE	FALSE	FALSE
Basel-Landschaft	BL	Liestal	518	289468	0	FALSE	FALSE	FALSE
Fribourg [Freiburg]	FR	Fribourg	1671	321783	0	TRUE	FALSE	FALSE
Valais [Wallis]	VS	Sion	5225	345525	0	TRUE	FALSE	FALSE
Ticino [Tessin]	TI	Bellinzona	2812	351491	0	TRUE	FALSE	FALSE
Luzern	LU	Luzern	1493	413120	0	TRUE	FALSE	FALSE
Genève [Genf]	GE	Genève	282	504128	0	FALSE	FALSE	FALSE
St Gallen	SG	St Gallen	2026	510734	0	TRUE	FALSE	FALSE
Aargau	AG	Aarau	1404	685845	0	TRUE	FALSE	FALSE
Vaud [Waadt]	VD	Lausanne	3212	805098	0	TRUE	FALSE	FALSE
Bern	BE	Bern	5941	1039474	0	TRUE	FALSE	FALSE
Zürich	ZH	Zürich	1729	1539275	0	TRUE	FALSE	FALSE

function: Canton -> Surface (every canton has a surface)

Not a function: Population -> Capital (population does not determine the canton)

injective function: Canton -> Capital (every canton has a different capital)

Not injective: Canton -> Population, different cantons may have same population

Week 5	Relations, Sequences, Summations
Week 6	Algorithms
Week 7	Complexity of Algorithms

Sorting the Table

Canton	Acronym	Capital	Surface	Population		Surface > 1000	Population < 100000	Surface > 1000 AND Population < 1000000
Zürich	ZH	Zürich	1729	1539275	0	TRUE	FALSE	FALSE
Bern	BE	Bern	5941	1039474	0	TRUE	FALSE	FALSE
Vaud [Waadt]	VD	Lausanne	3212	805098	0	TRUE	FALSE	FALSE
Aargau	AG	Aarau	1404	685845	0	TRUE	FALSE	FALSE
St Gallen	SG	St Gallen	2026	510734	0	TRUE	FALSE	FALSE
Genève [Genf]	GE	Genève	282	504128	0	FALSE	FALSE	FALSE
Luzern	LU	Luzern	1493	413120	0	TRUE	FALSE	FALSE
Ticino [Tessin]	TI	Bellinzona	2812	351491	0	TRUE	FALSE	FALSE
Valais [Wallis]	VS	Sion	5225	345525	0	TRUE	FALSE	FALSE
Fribourg [Freiburg]	FR	Fribourg	1671	321783	0	TRUE	FALSE	FALSE
Basel-Landschaft	BL	Liestal	518	289468	0	FALSE	FALSE	FALSE
Thurgau	TG	Frauenfeld	991	279547	0	FALSE	FALSE	FALSE
Solothurn	SO	Solothurn	791	275247	0	FALSE	FALSE	FALSE
Graubünden	GR	Chur	7105	199021	0	TRUE	FALSE	FALSE
Basel-Stadt	BS	Basel	37	195844	0	FALSE	FALSE	FALSE
Neuchâtel [Neuenburg]	NE	Neuchâtel	803	176496	0	FALSE	FALSE	FALSE
Schwyz	SZ	Schwyz	908	160480	0	FALSE	FALSE	FALSE
Zug	ZG	Zug	239	127642	0	FALSE	FALSE	FALSE
Schaffhausen	SH	Schaffhausen	298	82348	0	FALSE	TRUE	FALSE
Jura	JU	Delémont	836	73584	0	FALSE	TRUE	FALSE
Appenzell Ausserrhoden	AR	Herisau	243	55445	0	FALSE	TRUE	FALSE
Nidwalden	NW	Stans	276	43087	0	FALSE	TRUE	FALSE
Glarus	GL	Glarus	685	40590	0	FALSE	TRUE	FALSE
Obwalden	OW	Sarnen	490	37930	0	FALSE	TRUE	FALSE
Uri	UR	Altdorf	1077	36703	1	TRUE	TRUE	TRUE
Appenzell Innerrhoden	AI	Appenzell	173	16128	0	FALSE	TRUE	FALSE

How did the computer sort?

A simple **algorithm**

- Take the first
- Take the second, put it into right position
- Take the third, put it into right position
- Etc.

How much time? (worst case **complexity**)

$1 + 2 + 3 + \ldots + n = n*(n-1)/2$
a **sequence** and its **summation**

We can do much better: $n \log_2(n)$

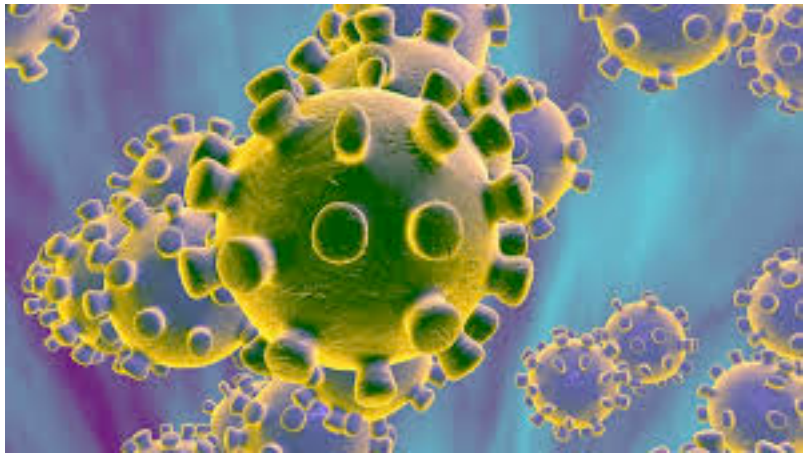
e.g. 1000 cities

n*(n-1)/2	approx. 500’000 steps
n log(n)	approx. 10’000 steps

Example 2: Covid Infections

One person has Covid-19.
A person can infect another person as from the second day after infection.
From then on, the person will infect every day another person.

How many persons are infected after n days, assuming that persons never recover from the disease?



Calculation

Day 1:

Day 2:

Day 3:

Day 4:

Day 5:

Day 6:

…

Day n: persons from Day n-1 (existing ones) plus persons from Day n-2 (who can infect people)

Week 5	Relations, Sequences, Summations
Week 9	Induction and Recursion
Week 11	Advanced Counting

Writing it down

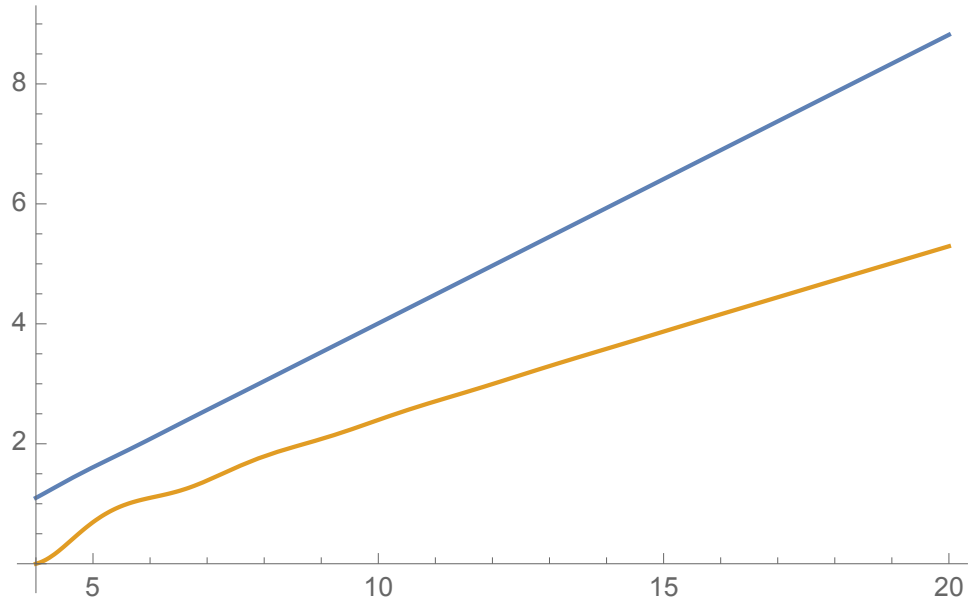
$f_n = f_{n-1} + f_{n-2}$, a **recursion**

But we can compute it directly (exponential growth)

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Introducing Quarantaine

How many persons are infected after n days, assuming that persons are quarantained after 4 days? $f_n = f_{n-1} + f_{n-2} - f_{n-4}$



Logarithmic scale

Example 3: Poker

How many poker hands of five cards can be dealt from a standard deck of 52 cards?

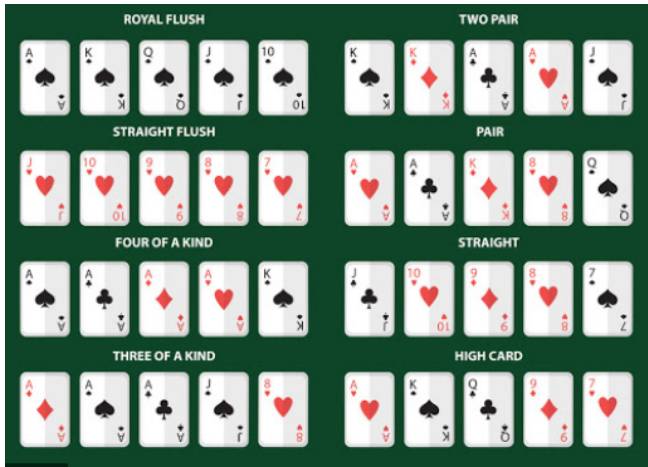
Simple: $n_1 = 52*51*50*49*48 = 311875200$

How many ways can you reorder five cards? (**permutations**)

Simple: $n_2 = 5*4*3*2*1 = 120$

How many different poker hands to exist (**combinations**)

$n_1 / n_2 = 2598960$



Week 10	Counting
Week 12	Probability

Full House

How many poker cards are full house (**counting**)?

13 kind of cards to select the three, e.g. Aces

4 ways to select three Aces (we have to skip one color)

12 kind of cards left for the pair

6 ways to select two cards out of 4 – (1,2)(1,3)(1,4)(2,3)(2,4)(3,4)

So in total $13 \cdot 4 \cdot 12 \cdot 6 = 3744$ ways to select a full house

What is the **probability** of a full house: approx. 0.15%

Week 10	Advanced Counting
Week 12	Advanced Probability

Poker

In how many ways can you change the order of 5 cards without leaving a card in the same position: 41

$$D_n = n!\left[1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!}\right].$$

How many deals do you need have more than 50% chance of having a full house: about 31

Example 4: Primes

A **prime** is a number that can be divided only by itself and 1

Examples: 1, 2, 3, 5, 7, 11, 13, 17, …. , $2^{82,589,933} - 1$, …..

24,862,048 digits (discovered December 2018)

Theorem: If n is an integer greater than 1, then n can be written as the product of primes (fundamental theorem of arithmetic)

Proof: not complicated but needs the principle of **induction**

Week 3	Proofs
Week 8	Number Theory
Week 9	Induction and Recursion

Proof

True for n = 2: since 2 is a prime, it is a product of primes (one factor)

Now assume true for all integers $2 \leq j \leq k$:

Two possibilities:

1. k+1 is prime, then k+1 is the product of (one) prime
2. k+1 is not prime.
   Then k+1 = a*b for two integers that are not 1 and not k+1.
   Therefore a and b are smaller or equal than k.
   But we have shown that then a and b can be written as product of primes.
   Therefore also k+1 is written as product of primes.

The principle of **(strong) induction** allows us now to conclude that the statement is true for any n.

Number of Primes

Euclid
(325 B.C.E. – 265 B.C.E.)

Theorem: There are infinitely many primes. (Euclid)

Proof Idea:

Assume there are only finitely many primes: $p_1, p_2, \ldots, p_n$

Let $q = p_1 * p_2 * \cdots * p_n + 1$

q would not be prime; using the fundamental theorem of arithmetic we can show that this leads to a contradiction (**proof by contradiction**)

Thus there are infinitely many primes.

Data, Life, Games & Numbers

Week 1	Propositional Logic
Week 2	Predicate Logic
Week 3	Proofs
Week 4	Sets, Functions, Relations
Week 5	Relations, Sequences, Summations
Week 6	Algorithms
Week 7	Complexity of Algorithms
Week 8	Number Theory
Week 9	Induction and Recursion
Week 10	Counting
Week 11	Advanced Counting
Week 12	Probability
Week 13	Advanced Probability