Podzimní soustředění KSP 2023 – Seznam přednášek

Tento spisek jest nabídkou přednášek, které byste na soustředění mohli slyšet, čili jakási obdoba matfyzácké Karolínky (ta je ale, pravda, ještě stále o něco tlustší). Přednášek je daleko víc, než kolik se dá za pár dní stihnout, a tak je na vás, abyste si vybrali, o které máte opravdu zájem. Pokud byste rádi slyšeli ještě o něčem dalším, klidně si o to napište (např. na Discord), třeba se najde někdo, kdo by vám o tom rád pověděl. Berte a vychutnávejte!

Údaje o jedné přednášce vypadají asi takto:

Stručný úvod do základů teorie vlkodlaků *("Za dne ukryt v hloubi lesa, děs temný zvečera se plazí...")* **LYK**
RNDr. Á. Cula
Úvod do moderní teorie vlkodlaků, čili též praktická dæmonologie a naiadologie.
Předpoklady: Měsíc v úplňku.

Dozvíte se (čteno v obvyklém pořadí): jméno přednášky, v uvozovkách motto přednášky, kód (pro snadnější odkazování na konkrétní předměty), jméno přednášejícího a nakonec stručný obsah přednášky. Hvězdičky znamenají obtížnost.

Základní přednášky

V této kategorii sídlí přednášky, které se dají považovat za základní stavební kameny informatiky, ať teoretické, či praktické.

Algoritmy a datové struktury

Základní algoritmy a jejich složitost *("Čím menší je časová složitost algoritmu, tím větší je složitost kódu.")* **ZAKL**
Pravděpodobně dvoudílná přednáška pro ty, kdo potřebují dohnat základní znalosti nutné pro ostatní přednášky. Zadefinujeme si základní pojmy jako je algoritmus, program, rekurze a jak se počítá jejich časová složitost, bude následovat přehled základních algoritmů – převážně třídění, rychlé hledání k-tého nejmenšího prvku, práce s výrazy a další.

Grafy & algoritmy *("Pojďme si hrát s obrázky.")* **GA**
Kiki Prokopová, Ján Plachý, Jirka Kalvoda, Vojta Káně
Co to jsou grafy, jak je v programech reprezentovat a hlavně k čemu se dají použít. Prohledávání grafu do šířky i do hloubky. Hledání nejkratších cest: Dijkstrův a Floydův algoritmus. Minimální kostry a Union-Find problem.

Těžké problémy * **HARD**
Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
V rámci této přednášky se budeme zabývat problémy tak těžkými, že nikdo na světě pro ně neumí vymyslet efektivní (rozuměj polynomiální) algoritmus. Spousta lidí dokonce věří, že to vůbec možné není. Abychom mezi tyto problémy pronikli, seznámíme se s pojmy NP-úplnosti a NP-těžkosti. Především si však konkrétní těžké úlohy ukážeme a naučíme se i některé těžké úlohy rozpoznat. Závěrem si řekneme, jak se s těžkými úlohami vypořádat v praxi.

Nejkratší a jiné cesty * *("Všechny cesty vedou do Horní Dolní, jen některé přes Řím.")* **CESTY**
Martin „Medvěd" Mareš, Jirka Kalvoda
O problému hledání cest v grafech trochu podrobněji. Obecné relaxační schéma, Bellmanův-Fordův a Dijkstrův algoritmus a jejich zrychlení pomocí různých datových struktur. Potenciálová redukce a heuristiky (třeba A^*), zaokrouhlování délek hran. Souvislosti s násobením matic: transitivní uzávěr, Seidelův algoritmus, Kleeneho algoritmus a regulární výrazy.

Toky v sítích *("Když je v grafu povodeň, těsní?")* **TOKY**
Ondra Sladký, Dan Skýpala, Říša Hladík, Michal Kodad, Jirka Kalvoda, Vojta Káně
K čemu je dobré, když grafem teče voda. Předvedeme si klasický problém toků v sítích a jeho všelijaké, mnohdy dosti překvapivé aplikace. Jak rozestavět n věží na šachovnici a jak ji místo toho pokrýt dominovými kostkami? Další souvislosti, jako třeba násobná souvislost grafů.
Předpoklady: Umět plavat (zejména v matematice)

Toky v sítích pro pokročilé * *("Když Edmons-Karp nestačí")* **TOKY2**
Říša Hladík, Jirka Kalvoda
Předvedeme si několik rychlejších algoritmů pro problém maximálního toku. Dinicův algoritmus a jeho mnohá vylepšení. Všelijaké příbuzné problémy: assignment problem, aneb hledání nejlevnějšího bipartitního párování. Maximální tok minimální ceny, aneb co když za průtok trubkami musíme platit? Ukážeme si algoritmus založený na postupném vylepšování a předvedeme si na něm obecnou myšlenku, kterou můžeme použít u optimalizačních problémů. Nahlédneme, že všechny zmíněné problémy můžeme popsat jako lineární programy, a proč se to občas vyplatí dělat. Pokud zbyde čas, řekneme si, co se změní, když budeme chtít vedle ropy v jedné síti zároveň přepravovat i čaj a Kofolu, a proč je to problém výrazně těžší, ale přesto řešitelný.
Předpoklady: TOKY

Datové struktury pro začátečníky *("Pole oraná a neoraná, stromy ovocné a okrasné.")* **DS1**
Lukáš Veškrna, Ján Plachý, Jirka Kalvoda, Vojta Káně
Jak si ukládat data natolik šikovně, abychom je nejen neztratili, ale také našli dříve, než si pro nás přijde Smrť. Klasické struktury jako pole, seznamy, fronta a zásobník, trie, vyhledávací stromy (vyvážené, AVL, *a-b*, splay), haldy (binární a obecně regulární) a v neposlední řadě hešování.

Datové struktury pro pokročilé * *("Haldy a jiné kupky.")* **DS2**
Ondra Sladký, Dan Skýpala, Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
Důmyslnější varianty vyhledávacích stromů: splay stromy, BB-α stromy, rankové stromy, vícerozměrné stromy. Chytřejší haldy: binomiální, Fibonacciho, rank-pairing. Amortizovaná analýza složitosti. Též několik přátelských randomizovaných datových struktur: skip listy a treapy.

Datové struktury pro ještě pokročilejší ** *("log log log log ... glo glo glo ...")* **DS3**
Ondra Sladký, Dan Skýpala, Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
Na přednášce si ukážeme některou z méně známých složitějších datových struktur. Pokud Ti ostatní přednášky přijdou moc jednoduché, tato je ta pravá pro Tebe.

Splay stromy *("Lepší než uklízení je organizovaný chaos.")* **SPLAY**
Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
Zapomeňte na pracné vyvažování vyhledávacích stromů. Místo toho zavedeme triviální pravidlo: pokaždé, když pracujeme s nějakým prvkem, vytáhneme ho do kořene stromu. Ukážeme, že toto pravidlo stačí na dosažení logaritmické složitosti, tedy aspoň amortizovaně. Také dokážeme, že Splay strom je nejhůře konstanta-krát horší než libovolný jiný strom, a možná i spousta dalších magických vlastností.

Stromové algoritmy *("Půjdeme na to od lesa")* **TREES**
Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
Stromy jsou jednou z nejtypičtějších (a nejjednodušších) odrůd grafů. Ledacos pro ně umíme řešit mnohem rychleji než pro obecné grafy, tak se pojďme podívat, jak se to dělá. Předvedeme několik obecných technik pro práci se stromy: DFS očíslování, „vandalskou indukci", intervalové reprezentace. Různé rozklady: heavy-light, Fredericksonův, separátorový a ST-stromy.

Magické algoritmy * *("Pokročilá magie není rozlišitelná od technologie.")* **MAGIC**
Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
O algoritmech značně magických a nečekaných. Jak násobit n-ciferná čísla rychleji než v kvadratickém čase. Kouzlo na slévání setříděných posloupností v konstantním prostoru. Isomorfismus stromů pomocí přihrádkového třídění. Bitové kejklířství. Hledání největší díry.

Persistentní datové struktury * *("Datové struktury cestující časem.")* **PERS**
Ondra Sladký, Dan Skýpala, Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
Ukážeme si (téměř) obecný způsob, jak naučit datové struktury zapamatovat si celou svou historii. Předvedeme si, jak tuto historii modifikovat a k čemu to je celé dobré.

Aproximační algoritmy *("Součet úhlů v trojúhelníku je vždycky tři ... tedy alespoň $\pm 5\,\%$.")* **APX**
Říša Hladík, Jirka Kalvoda
Některé úlohy jsou tak těžké, že je za dobu existence tohoto vesmíru nedokážeme vyřešit (naneštěstí to zatím o většině takových úloh ani nesvedeme dokázat). Co kdyby nám ale stačilo i řešení, které je nejhůře o $10\,\%$ horší, než to optimální? Ukážeme si pár klasických aproximačních algoritmů a aproximačních schémat: 2-aproximace bin packingu, obchodní cestující ve 2D, množinové pokrytí, MaxSAT a další.

Beyond worst-case datové struktury * *("Třídit rychleji než v $\Theta(n \log n)$ porovnání nejde... ledaže...")* **BEYWORST**
Říša Hladík
Je známým faktem, že vložit n prvků do haldy nebo vyhledávacího stromu a pak je od nejmenšího vybrat obecně nejde rychleji než v čase $\Theta(n \log n)$, tedy $\Theta(\log n)$ na operaci. Co když je ale posloupnost operací speciální – třeba proto, že je vykonává Dijkstrův algoritmus, nebo proto, že vkládáme téměř setříděné pole? Ukážeme si haldu ostře lepší než tu Fibonacciho, jejíž složitost závisí na operacích a může být až konstantní, a dokážeme, že Dijkstrův algoritmus s touto haldu je optimální.
Předpoklady: základy amortizace; znalost Fibonacciho hald není potřeba

Intervalové stromy * *("Já bych ty intervaly nejradši... dal do stromu!")* **ITREE**
Říša Hladík, Jirka Kalvoda
Intervalový strom je datová struktura pracující s intervaly, se kterou se můžeme setkat v mnoha úlohách (zejména soutěžních). Řekneme si, co to intervalový strom je, jaké všechny druhy intervalových stromů existují a jejich použití si ukážeme na úlohách. Na závěr si představíme jednu „magickou" datovou strukturu jménem Fenwickův strom.

Dynamické programování *("Kampak jsem si to jenom schoval?")* **DYNP**
Říša Hladík, Michal Kodad, Martin „Medvěd" Mareš, Kiki Prokopová, Ján Plachý, Jirka Kalvoda
Dynamické programování je programátorská technika využívající velice prostinkého nápadu: Proč něco počítat několikrát, když to mohu spočítat jednou a výsledek si uložit? Na této přednášce si ukážeme, že tento jednoduchý nápad může pomoci efektivně vyřešit i poměrně obtížné úlohy.

Hledání v textu *("»Vyšíváme v seníku!« – kde jsem to jen viděl?")* **TEXT**
Dan Skýpala, Říša Hladík, Michal Kodad, Honza Černý, Kiki Prokopová, Jirka Kalvoda
Někdy potřebujeme najít podřetězec ve velkém množství textu. Stromeček trochu připomínající ten biologický aneb trie. Proč se ve vstupu vracet neboli Knuthův-Morrisův-Prattův algoritmus. Hledání více řetězců najednou podle Aha a Corasickové. Okénkové hešování Rabina a Karpa.

Amortizace *("Celek bývá daleko menší než součet částí.")* **AMORT**
Říša Hladík, Martin „Medvěd" Mareš, Jirka Kalvoda
Spousta algoritmů je mnohem rychlejší, než jak na první pohled vypadají. Šikovný způsob, jak takové chování zkoumat, je amortizovaná časová složitost. Předvedeme několik trochu překvapivých příkladů amortizace: dvojková a jiná počitadla, datové struktury založené na přebudovávání, vyhledávací stromy bez otravného vyvažování, dynamizace datových struktur, udržování historie.

Distribuované počítání *("Víc hlav víc ví... teda, pokud mezi nimi vedou dostatečně krátké kabely.")* **DIST**
Říša Hladík, Jirka Kalvoda
Teoretická přednáška o tom, co se stane, když do vrcholů grafu umístíme počítače a necháme je, ať si mezi sebou povídají. Nějaké problémy, jako například barvení grafu, umíme v takovém modelu vyřešit rychleji než logaritmicky – a to přesto, že poslat zprávu z jednoho konce grafu na druhý trvá až lineárně. Co když navíc omezíme velikost zpráv, aneb LOCAL vs. CONGEST. Rychlé barvení ve stromě i obecně, maximální nezávislá množina, minimální kostry. Problémy na úplném grafu: routování, třídění.

Programovací jazyky a nástroje

Programování v jazyce C **C**
Martin „Medvěd" Mareš, Jirka Kalvoda, Vojta Káně
Jazyk C patří k nejrozšířenějším jazykům, hodí se pro low-level programování i kusy kódu, které mají zejména být rychlé. Představíme si datové typy a běžné programové konstrukce, vysvětlíme si základy práce s ukazateli a také se seznámíme se standardními knihovnami jazyka C.

Pokročilé povídání o Pythonu *("`import antigravity`")* **PYTH2**
Martin „Medvěd" Mareš, Jirka Kalvoda
Povídání o méně známých částech Pythonu. Datový model: objekty, třídy, metatřídy, dekorátory a deskriptory. Magické metody a na nich postavené protokoly. Generátory, generátorové výrazy a funkcionální styl programování. Asynchronní a paralelní programování. Zajímavé moduly nejen ze standardní knihovny. Propojení Pythonu s C, ...
Předpoklady: Základy Pythonu.

(Meta)programování v LISPu *("Průvan ve skladišti závorek.")* **LISP**
Martin „Medvěd" Mareš
Jak vypadá programovací jazyk z roku 1960, který je velmi jednoduchý, ale přitom tak mocný, že do něj skoro každou vymoženost moderních programovacích jazyků někdo dodělal jako knihovnu. Datový model tvořený atomy a krabičkami, z nichž stavíme seznamy a stromy. Kód je také druh dat: funkce vyššího řádu, makra, metaprogramování. Dialekty LISPu: Common LISP, Scheme a třeba také Clojure.

Procesy, vlákna a zámky * *("Koupil jsem dalších 15 procesorů, proč je to stále stejně pomalé?")* **THREAD**
Jirka Kalvoda, David Klement
Zrychlovat procesory už moc neumíme, tak si jich pořídíme více. Jak psát programy, které běží paralelně ve více procesech nebo vláknech. Jak vlákna usměrnit, aby nám nerozbila program na nečekaných místech. Rozebereme, jak fungují zámky, kdy je musíme použít a jakou cenu za to platíme.

Git a jiné systémy pro správu verzí *("U svatýho tučňáka, kdo sem napsal tohle? Ono to tvrdí, že JÁ?!")* **GIT**
Jan Černohorský, Jirka Kalvoda, Vojta Káně
Jak vyvíjet program delší dobu a nezbláznit se u toho. Různé systémy pro správu verzí od diff/patch přes CVS a SVN až ke Gitu. Jak Git funguje: stromy, commity, větve, tagy. Merge mezi větvemi nebo mezi různými počítači.

Git pro pokročilé *("In case of fire, commit, push, and exit the building.")* **GIT2**
Martin „Medvěd" Mareš, Jirka Kalvoda, Vojta Káně
Používáte Git pro všechny své programy a k svačině místo novin čtete commit logy svých oblíbených projektů? V tom případě pojďme nahlédnout pod pokličku, jak Git funguje uvnitř. Reprezentace historie pomocí hešování grafů. Pracovní strom, index, commity a jejich adresy, větve. Pack files jako elegantní způsob komprese dat na disku i na síti. Kouzelnické triky: hledáme bugy půlením historie, přepisujeme dějiny, automaticky konvertujeme soubory. Git v praxi: jak se liší správa zdrojáků v projektech o jednom, deseti a tisíci programátorech. Udržujeme patche k cizímu programu aneb StGit.

Jak se nestat vepřem *("/* You are not expected to understand this */")* **STYLE**
Martin „Medvěd" Mareš, Jirka Kalvoda, David Klement
Tvrdí se, že číst kód je mnohdy těžší, než ho psát – dokonce i po sobě, stačí krátká doba. Je několik obecně uznávaných pravidel, jak kód psát a jak ne, aby byl hezký a dobře čitelný. Od základních (rozumná pojmenovací konvence, systematické odsazování), až po to, kdy opravdu použít `goto`, jak členit program na funkce a jak využít nějaké třídy, moduly a podobně. Jak napsat užitečný komentář nebo dokumentaci. A kdy se vyplatí se na všechna tato pravidla vybodnout.

Visual Studio Code **VSCODE**
David Klement
Visual Studio Code je jednak univerzálním textovým editorem, jednak mocným vývojovým prostředím. Proč mít na každý jazyk jiný program, když VS Code stačí na všechno? Kromě editoru samotného si ukážeme, co všechno umí automatizovat a jak v něm vypadá efektivní workflow.

Textový editor Vim *("Víš, jaký je nejlepší textový editor? Vim.")* **VIM**
Martin „Medvěd" Mareš, Jirka Kalvoda
Odložme na chvíli své myše a pojďme si vyzkoušet textový editor, který umí poslouchat na slovo. Pravda, budeme se ta slova muset chvíli učit, ale výsledek bude proklatě efektivní. Základní příkazy, textové objekty, regulární výrazy, makra, kouzla. Neovim, Lua a language servery. Vimovité ovládání jiných programů, třeba webového prohlížeče.

Ladicí nástroje * *("Jak se ladí kytara, jak křišťálová koule a jak program.")* **GDB**
Martin „Medvěd" Mareš, Jirka Kalvoda, Vojta Káně
Kdo píše programy, které vždy hned fungují, ať se přihlásí. A kdo ne, ať se přihlásí na tuto přednášku. Ukážeme si různé (především nízkoúrovňové) ladicí nástroje a techniky: debugger (neboli odvšivovač) gdb, strace, a valgrind. Kdy je použít a kdy se více hodí `printf` a `assert`. Pokročilejší kouzla: reverzibilní debugger `rr`, profilování pomocí `perf`u a `bpftrace`.
Předpoklady: Znalost nějakého nízkoúrovňového jazyka (třeba C).

SQL databáze *("`SELECT something FROM knowledge LIMIT 90min`")* **SQL**
Martin „Medvěd" Mareš, Jirka Kalvoda, Vojta Káně
Jak si schovat data do relační databáze a jak je tam zase najít, ideálně rychle. Definice tabulek a indexů. Dotazy a jejich skládání a vnořování. Pohledy, funkce a triggery. Transakce a různé druhy konzistence. Rozdíly mezi dialekty SQL.

Hardware a operační systémy

Správa paměti * *("Když má program sklerózu...")* **MEM**
David Klement, Jirka Kalvoda, Vojta Káně
Po chvíli zjistíme, že nám lokální a globální proměnné nestačí a je potřeba paměť alokovat dynamicky. Co všechno si musíme udělat sami a co se děje programátorovi „za zády". Mapování adresního prostoru, ruční alokování a vracení paměti a problémy s tím spojené (chyby programátora), počítání odkazů a daň s nimi spojená (a hele, cyklus), odklízeče odpadu (mark & sweep, kopírovací, generační a jiné triky).

Principy počítačů *("A opravdu uvnitř počítače běhají malí trpaslíci?")* **HW**
Martin „Medvěd" Mareš, Jirka Kalvoda
Vydáme se do země skřítků, kteří pohánějí počítače. Počítačové architektury od hodinek po superpočítač od Craye, jejich křivolaká historie i současnost. Co je to procesor, jak se programuje a jak se chová. Různé druhy pamětí a jejich cacheování. Jak procesory komunikují s okolím – sběrnice, čipové sady, vstupní a výstupní zařízení. A co když je procesorů několik, nebo třeba pár tisíc? Přednáška bude praktická: pár počítačů při ní rozebereme a možná i nějaký postavíme.

Bezpečnostní chyby v procesorech *("Sběrnicí obchází Přízrak a krade klíče.")* **CPUBUG**
Martin „Medvěd" Mareš, Jirka Kalvoda
Že jsou v programech bezpečnostní chyby, na to jsme si už zvykli. Ale teprve zvolna si zvykáme na to, že mohou být i v hardwaru, dokonce v samotném procesoru. Nedávné roky přinesly několik ošklivých překvapení tohoto druhu s veselými jmény, jako je Meltdown a Spectre. Budeme se zabývat fungováním procesoru uvnitř, zejména všelijakými triky na zrychlení výpočtu: superskalárním zpracováním instrukcí, kešováním a predikcí skoků. A ukážeme, co pokazil Intel, co AMD a jak toho jde zneužít.

Operační systémy *("Mám 3GHz procesor, tak co to už půl hodiny dělá?")* **OS**
Jirka Kalvoda, Vojta Káně
Jak vypadá architektura dnešních operačních systémů aneb co všechno musí systém zařídit, aby na něm programy fungovaly. Správa procesů a vláken, plánování, synchronizace. Paměť, adresace a její přidělování. Správa souborů, filesystémy. Čemu se říká jádro a proč se spojuje s pudlem.

Toulky assemblerem *("Pojďme se společně ztratit.")* **ASM**
David Klement, Martin „Medvěd" Mareš
Procesor nerozumí proměnným, podmínkám, cyklům ani jiným věcem, které jsou pro nás při programování běžné. Podíváme se, jak lze výše zmíněné konstrukce přepsat do procesorových instrukcí. Co přesně se děje při volání funkce? Jak kód vytunit, aby běžel rychleji? Také se podíváme na souvislosti s návrhem dnešních procesorů.
Předpoklady: Umět přečíst jednoduchý program v C.

Cache-oblivious algoritmy *("Kešuješ, kešuje, kešujeme.")* **CACHE**
Martin „Medvěd" Mareš, Jirka Kalvoda
Dnešní procesory mají několik úrovní vyrovnávacích pamětí (cache), což způsobuje, že ačkoliv si jsou všechny části paměti rovny, některé si jsou rovnější. Jak taková cache funguje? Jak se procesor rozhodne, co si v ní zapamatuje a co vyhodí? Jak toho můžeme využívat při programování, aby naše programy běžely rychleji? Předvedeme kousek teorie i několik praktických ukázek s poněkud překvapivým chováním.
Předpoklady: Kešu oříšky

Programování v Linuxu *("Všechno na světě je tak trochu soubor.")* **PLX**
Martin „Medvěd" Mareš, Jirka Kalvoda, Vojta Káně
Jak vypadá rozhraní mezi jádrem Linux a uživatelskými programy. Co se doopravdy stane, pokud ve svém céčkovém programu zavoláme `printf` nebo `malloc`. Jak napsat program, který vůbec nepotřebuje standardní céčkovou knihovnu. Co všechno se umí chovat jako soubor a co jako signál.
Předpoklady: Schopnost přečíst a napsat jednoduchý program v C.

Zápisky ze správy Windows *("„Please contact your system administrator." A co když jsem to já?!?")* **WIN**
Vojta Káně, Jan Černohorský
Když se zatvrzelý Linuxák vydá na druhou stranu barikády, je to tvrdé, ale ne nepřínosné. V koláži vlastních zkušeností ukážu, kde se dá u Redmondu inspirovat a kde raději ne. Podíváme se na Active Direcory i její reimplementaci v podobě Samby. A podumáme, jak nesmiřitelné světy zkřížit pro náš maximální prospěch za cenu ztráty politických bodů na obou stranách.

SystemD *("/etc/systemd/annotation/override.d/50-SKSP.conf")* **SYSTEMD**
Vojta Káně
Navzdory plamenný internetovým debatám, považuji tento široký balík démonků za velmi mocný a užitečný. Začneme s initem, napíšeme pár služeb a prozkoumáme, jak je konfigurovat. Z toho pochopíme, jak psát a komponovat konfiguraci ostatních démonů, a na závěr projdeme i méně známé z nich.
Předpoklady: Základní znalost Linuxu

Mikrokontroléry *("Nejlepší debugger je LEDka.")* **MCU**
Martin „Medvěd" Mareš
Srdcem mnoha dnešních technických hraček je mikrokontrolér. To je čip, na kterém je integrovaný nejen procesor, ale i paměť a spousta zajímavých periferií. Ukážeme si, jak se mikrokontroléry programují, jaké periferie typicky obsahují a jak je používat ke komunikaci s okolním světem. Něco si vyzkoušíme i prakticky na STM32.
Předpoklady: Hodí se základní znalost jazyka C.

Linux pro správce serveru *("Printer is on fire???")* **LSERV**
Martin „Medvěd" Mareš, Jan Černohorský, Jirka Kalvoda
Jak vytvořit jednoduchý Linuxový server, který poskytuje služby vaší domácnosti, nebo třeba nějaké větší síti. Co se tam hodí provozovat? Povíme o SSH, kličích, šifrování, systemd, Apache a Nginxu, nastavení mailového serveru i DNS. Jak server zabezpečit před útočníky, jak před ztrátou dat a jak před uklízečkou. Vše si vyzkoušíme prakticky, třeba na virtuálním počitači.
Předpoklady: Základní znalost Linuxu.

Sítě a bezpečnost

Sítě a Internet *("Sítě nejen na ryby.")* **NET**
Martin „Medvěd" Mareš, Jan Černohorský, Jirka Kalvoda, Vojta Káně
Jak funguje Internet a počítačové sítě vůbec: od elektronů v drátech (fotonů v optických kabelech nebo elektromagnetických vln) přes packety a jejich forwarding až k jednotlivým síťovým službám. Adresace, internetworking a dynamický routing. Jak NAT zachránil i zničil Internet a proč se těšíme na IPv6.

Sítě II – protokoly a síťové útoky *("Jak si přečíst maily... sousedovy maily.")* **NET2**
Martin „Medvěd" Mareš, Jan Černohorský, Jirka Kalvoda, Vojta Káně
Volné navázání na NET. Budeme si povídat o tom, co za data nám po síti běhá a jaké se k tomu používají protokoly – DNS, FTP, HTTP nebo třeba i mailové SMTP a IMAP. Zaměříme se více na ty nejpoužívanější (metody GET a POST v HTTP), nakousneme cacheování a nadlábneme se cookies. A pokud zbude čas, využijeme zranitelnosti některých protokolů a provedeme síťový útok.
Předpoklady: Základní povědomí o počítačových sítích

Webové stránky **WWW**
Jan Černohorský, Jirka Kalvoda, Vojta Káně
Co se děje za oponou, když do prohlížeče zadáte adresu svých oblíbených stránek? A jak si takovou stránku taky pořídit? Přelet nad protokolem HTTP, seznámení s HTML a předvedení kaskádových stylů. Jak fungují dynamické stránky od formulářů až po JavaScript běžící v prohlížeči.

E-mail *("Drahoušek zákazník.")* **EMAIL**
Jirka Kalvoda, Vojta Káně
Co se stane s e-mailem, když jej odešlete? Kudy chodí a kudy jej čerti nesou? Jaké máte záruky, že přijde; proč občas přijde pozdě nebo vůbec. Problém formátů a kódování, chyby webových i jiných klientů. Protokoly SMTP, POP, IMAP a co se stane, když do nich přimícháme SSL/TLS. E-mailová bezpečnost, SPAM a (nefunkční?) obrana pomocí SPF, DKIM a DMARC. Nakonec se podíváme na ne zrovna triviální grafový problém, který je v emailech skrytý.

Kryptografie *("Gbgb arav zbp gnwan mcenin.")* **CRYPT**
Martin „Medvěd" Mareš, Jirka Kalvoda
Kryptografie čili tajuplná nauka o šifrách, jejich konstrukci a hlavně o jejich luštění. Šifrovací systémy jako lego: základními kostičkami nám budou symetrické a asymetrické šifry, jednosměrné funkce a náhodné generátory. Stavět z nich budeme kryptografické protokoly na bezpečný přenos, autentikaci, digitální podpisy a třeba i na házení korunou po telefonu. Předvedeme nerozluštitelnou šifru a dokonce to o ní i dokážeme.

Aplikace kryptografie * *("6140 a184 c9a6 41f1 de99 e733 354a f451")* **CRYPT2**
Martin „Medvěd" Mareš, Jirka Kalvoda
Pokročilejší a občas nečekané aplikace základních kryptografických primitiv. Jak přesvědčit server, že známe heslo, aniž bychom mu ho posílali? Jak zajistit, aby útočník nemohl dešifrovat komunikaci, ani když dodatečně získá soukromý klíč? Jak funguje BitCoin (decentralizovaná digitální měna) či Tor (protokol znemožňující komukoli po cestě vědět, kdo s kým komunikuje)?
Předpoklady: Základní povědomí o šifrování (CRYPT) a víra v existenci náhodných čísel

Praktická kryptografie *("A proč jsou všechny ty zámky na papírových dveřích?")* **PCRYPT**
Martin „Medvěd" Mareš, Jirka Kalvoda
Programátoři si často myslí, že pro bezpečnou komunikaci stačí vybrat si z knihovny osvědčenou silnou šifru. Jak naivní! Navrhnout bezpečný protokol není maličkost a dá se při tom ledacos zpackat. Replay útoky (jak otevřít auto krabičkou za 30 dolarů), útoky na padding a na blokovou strukturu. Čí že je ten podpis? Jak nepoužívat RSA a jak nehešovat hesla. Jak náhodná jsou vaše čísla? Postranní kanály: časování, spotřeba, záření. K čemu se crackerům hodí termoska s tekutým dusíkem.

Kryptografie na eliptických křivkách ** **ECC**
Ondra Chwiedziuk
O eliptických křivkách se v asymetrické kryptografii velice často mluví. Říká se o nich, že jsou bezpečnější než RSA, ale nejsou bezpečné vůči útokům kvantových počítačů. Proč tomu tak je? Jak takové šifrování funguje? Co vlastně jsou ty eliptické křivky? Všechny tyhle otázky se pokusí zodpovědět tato přednáška. Před přednáškou se hodí vědět základy kryptografie a RSA.

Umělá inteligence

Přírodou inspirované algoritmy * **PRINSALG**
Honza Černý
Příroda je nádherná a celá tisíciletí se jí inspirujeme. Kolik už inspirovalo spisovatelů, básníků a malířů. Nyní jsou na řadě programátoři. Když si nebudeme vědět s nějakým těžkým problémem rady, tak zkusíme nenápadně opisovat od přírody.

Umělá inteligence * **AI**
Michal Kodad, Honza Černý
Ukážeme si, jak počítače přemýšlí při řešení problémů a jakým způsobem hledají řešení. Volně se dostaneme k prohledávání stavového prostoru (který bývá exponenciálně velký) a ukážeme si různé jak informované, tak neinformované techniky pro jeho procházení. Setkáme se třeba s algoritmy, které jsou použity v GPS.

Herní algoritmy *("Když nemáte na to, abyste vyhráli šachový turnaj...")* **AIGAME**
Michal Kodad
Povídání o tom, jak programovat počítačové soupeře do šachů a her jim podobným. Základní minimaxový algoritmus a jeho vylepšení neboli α-β ořezávání. Stále pomalé? Několik nápadů na efektivnější ořezávání. Ne u všech her však funguje hrubá síla (minimax) dobře, ukážeme tedy ještě pravděpodobnostní přístup Monte Carlo Tree Search.

Strojové učení *("Nechme stroje se samy učit.")* **ML**
Michal Kodad
Co je to strojové učení? Jaké typy strojového učení existují? Začneme u jednoduché lineární regrese, přes perceptron až skončíme u kouzelného slovíčka neuronové sítě. Povíme si rozdílné druhy neuronových sítí a nakonec si odskočíme k algoritmu, který nepotřebuje kromě surových dat nic navíc a dokáže dělat užitečné věci.

Neuronové sítě **NEURO**
Ondra Sladký, Michal Kodad
Základy strojového učení – od lineární a logistické regrese přes husté neuronové sítě až po konvoluční neuronové sítě, které se používají při rozpoznávání obrázků. Během přednášky si ukážeme i nějaké paralely s neurologií.

Úvod do počítačové linvistiky *("Petr si koupil vstupenku. Vsunul ji do kapsy. Byla děravá.")* **PCLING**
Honza Černý
Zahrajeme si na tlumočníky mezi stroji a lidmi. Ukážeme si různé reprezentace lidského jazyka ve světě jedniček nul a ukážeme si základní algoritmy, co s nimi pracují.

Transformery ** **NEURO2**

Ondra Sladký, Michal Kodad

Tato navazující přednáška na Neuronové sítě se zabývá transformery, což je pokročilý model pro zpracování přirozeného jazyka. Transformery se staly revolučním nástrojem v oblasti strojového učení a jsou základem mnoha moderních aplikací, jako jsou strojový překlad, rozpoznávání řeči nebo generování textu. Během přednášky se seznámíte se základními principy transformerů, jako je self-attention mechanismus a enkodér-dekodér architektura. Budeme také diskutovat o jejich výhodách a omezeních a představíme některé příklady jejich úspěšného využití v různých oblastech. – Anotaci napsal ChatGPT

Grafika a typografie

Typografie *("What You See Is all What You've Got!?")* **TYPO**

Martin „Medvěd" Mareš, Jirka Kalvoda

Jak na počítači text nejen napsat, ale také vysázet tak, aby pěkně vypadal a aby (což je důležitější) se i příjemně četl. Jak se sází pohádka, jak báseň a jak vzorové řešení KSP plné komplikovaných vzorců. Jak jde dohromady staleté umění typografické a moderní technika. Přineste knihy i letáky, zkritizujeme sazeče, co se do nich vejde.

TEX *("No pages of output. Ask a TEXnician.")* **TEX**

Martin „Medvěd" Mareš, Jirka Kalvoda

Z předchozí přednášky máme představu o tom, jak vypadá pěkná sazba. K její výrobě nám pomůže typografický systém TEX. Praktická přednáška s ukázkami použití TEXu od hladké sazby knihy až po zběsilosti hraničící s programováním. Jak do TEXu vkládat obrázky a jak to raději nedělat. Kde shánět další informace: TEXbook, TEXbook naruby a další zajímavá literatura. Praktické rozdíly mezi různými dialekty TEXu. Všelijaká rozšíření: pdfTEX, eTEX, LuaTEX.

TEXnické detaily ** *("TEX capacity exceeded. Ask a wizard to enlarge me.")* **TEX2**

Martin „Medvěd" Mareš, Jirka Kalvoda

Pokročilejší přednáška o TEXu pro ty, kdo ho už nějaký čas používají. Budeme v TeXu programovat, kreslit obrázky, otáčet text, používat různé podivné fonty a třeba si i vysázíme odstavec ve tvaru kolečka.

Asymptote *("Vy obrázky kreslíte? My je programujeme!")* **ASY**

Martin „Medvěd" Mareš, Jirka Kalvoda

Rádi byste své řešení KSP ozdobili hezkými obrázky? Dají se nakreslit ručně, ale často je snazší obrázky programovat. Předvedeme Asymptote, což je programovací jazyk určený na kreslení 2D a 3D obrázků. Také se zastavíme u jeho předchůdců MetaPostu a MetaFontu a knihovny pro vektorové kreslení Cairo.

Formát PDF **PDF**

Martin „Medvěd" Mareš, Jan Černohorský

Jeden z nejrozšířenějších formátů na předávání dokumentů má za sebou spletitou historii i dokumentaci. Ukážeme si, jak vypadá uvnitř a co se do něj dá uložit: grafické objekty, text, fonty, odkazy, všelijaké anotace a meta-data, a dokonce i kryptografické podpisy. Zmíníme se o profilech, třeba PDF/X a PDF/A. Při troše štěstí si vytvoříme jednoduchý PDF soubor ručně a možná půjde i otevřít.

Unicode *("Jaký kód má sněhulák s kudrnatými vlasy?")* **UNI**

Martin „Medvěd" Mareš, Jan Černohorský

Jak funguje znaková sada Unicode, která se snaží zapsat všechny jazyky světa? Codepointy versus glyfy. Kombinující znaky, čtvero normálních forem a pátá lehce nenormální. Typografické a neviditelné znaky. Co všechno prozradí Unicode Character Database. Uložení v paměti: formáty UCS-2, UCS-4, UTF-8 a UTF-16, nešvar s BOM. Tajemný svět emoji. Jak se s Unicode programuje? A jako vždy: bezpečnostní problémy.

Obrázky **IMG**

David Klement

Jak zařídit, aby fotka nezabírala 40 MB? Povíme si o typických formátech PNG a JPEG, prozkoumáme, kdy se hodí který, a podíváme se jim pod pokličku. Naučíme se vymýšlet nové pixely při zvětšování obrázků. Také nakousneme vektorové obrázky a Bézierovy křivky.

Teoretická informatika

Pravděpodobnostní algoritmy *("Kudy dál? Hoďme si kostkou!")* **PPALG**

Martin „Medvěd" Mareš, Říša Hladík, Jirka Kalvoda

Když nevíme, jak se v algoritmu rozhodnout, někdy pomůže ponechat to náhodě a prostě si „hodit kostkou". Dokážeme sestrojit algoritmy, které jsou rychlé, i když spravný výsledek vydají jen v 99 % případů. Ale i takové, které odpoví správně vždycky, ale rychlé jsou jen v průměru (třeba QuickSort). Též ukážeme, jak pomocí náhody zabraňovat kolizím v hešování.

Kvantové počitání ** *("`return 0.5*dead + 0.5*alive;`")* **QC**

Říša Hladík, Martin „Medvěd" Mareš

Stručný úvod do kvantového počítání. Kvantová superpozice stavů výpočtu a její kolaps při měření. Základní kvantové operace: negace, řízená negace, permutace, Hadamardovo hradlo, Tofolliho hradlo. Kvantová teleportace a jakto, že není v rozporu s teorií relativity. Groverův algoritmus na hledání v odmocninovém čase. Kvantová Fourierova transformace a Shorův algoritmus pro faktorizaci.

Předpoklady: Znalost komplexních čísel je nutností, znalost lineární algebry výhodou.

Jazyky, gramatiky a automaty * *("Existuje regex, který rozpoznává regexy?")* **AUTO**
Říša Hladík, Honza Černý, Martin „Medvěd" Mareš
O jazycích přirozených, programovacích a matematických, jejich popisu a rozpoznávání. Začneme těmi nejjednoduššími: regulární jazyky a výrazy, konečné deterministické a nedeterministické automaty. Pak budeme stoupat po příčkách Chomského hierarchie, kam až to půjde. Jak výpočetně silný je třeba takový automat na kafe?

Modely počítačů *("Nač Pentium? Máme Turingovy stroje!")* **MODEL**
Martin „Medvěd" Mareš, Jirka Kalvoda
V HW se dozvíte, jak fungují „opravdové" počítače, zde pro změnu na čem počítají teoretici. Všechny počítače jsou si rovny, jen některé jsou si rovnější. Turingův stroj obyčejný, vícepáskový, nedeterministický a univerzální. Random Access Machine (RAM) a Pointer Machine. Trocha minimalismu aneb stroj s počítadly. Až nám začne být smutno, pořídíme si klidně N^2 procesorů a spřáhneme je do paralelního počítače (PRAM). Rychlé paralelní slévání a třídění. Pokud zbude čas, ukážeme si buněčné a grafové automaty, nebo třeba dlaždičky v koupelně.

Buněčné automaty a Game of Life *("Čtverečkovaný svět, co není Minecraft")* **LIFE**
Martin „Medvěd" Mareš, Ján Plachý
Game of Life je dvojrozměrný svět, ve kterém se buňky vyvíjí podle průzračně jednoduchých pravidel. Už desítky let v tomto světě objevujeme další a další zajímavé jevy. Tak do něj také nahlédneme, prozkoumáme souvislosti s evoluční biologií i s algoritmy. Též uvidíme, jak Život zapadá do obecnějšího světa buněčných automatů.

Složitější složitost ** *("Kolik sekund stojí jeden bajt?")* **SLOZ**
Martin „Medvěd" Mareš, Jirka Kalvoda
Teorie výpočetní složitosti opravdu důkladně. Různé definice výpočetního modelu a velikosti vstupu. Složitostní třídy a vztahy mezi nimi. Různé druhy redukcí. Třídy P, NP, L, NL, PSPACE a NPSPACE. Nedeterministické stroje, orákula, alternující stroje a polynomiální hierarchie. Neuniformní složitost.

Dolní odhady složitosti * **ODHADY**
Martin „Medvěd" Mareš
Jak dokázat, že jsme našli nejrychlejší možný algoritmus na danou úlohu? To je docela těžká otázka, ale aspoň u několika úloh na ni dokážeme najít odpověď. Nutnost přečíst celý vstup (opravdu?). Měříme množství informace: vážení kuliček, hledání, třídění a různost prvků v porovnávacím modelu. Omezená paměť a princip holubníku: závorkování. Komunikační složitost: palindromy na jednopáskových strojích. Nekonstruktivní důkazy existence těžkých problémů.

Aplikace informatiky

Bioinformatika **ACGT**
Ondra Sladký, Ján Plachý
Stručný úvod do bioinformatiky. Ukážeme si základní algoritmy na local a global alignment (BLAST, Smith-Waterman a jejich modifikace). Dále se zaměříme na algoritmy na sestavování referenčních genomů z jednotlivých readů a ukážeme si některé moderní algoritmy založené na k-merových metodách.

Komprese dat *("Jnm idln kpln j nstlčtln.")* **ZIP**
Martin „Medvěd" Mareš
Pokud jsou data příliš velká, můžeme je zkusit zkomprimovat. Předvedeme základní kompresní algoritmy: triviální (RLE), slovníkové (LZ77), statistické (Huffmanovo a aritmetické kódování) a některé pokročilejší techniky, jako třeba Burrowsovu-Wheelorovu transformaci (BZIP). Zmíníme se o kompresi zvuku, obrazu a videa (prediktory, wavelety, všelijaká ztrátová komprese).

Zpracování dat *("Bez práce nejsou koláč... ové grafy.")* **DATA**
Martin „Medvěd" Mareš
O světě jde sehnat spousta zajímavých dat ve strojově zpracovatelné podobě: obce a domy v nich, linky hromadné dopravy, katalogy hvězd, slova v češtině, katalog pokémonů, ... Pojďme se podívat, jak s daty zacházet. Naučíme se číst různé formáty dat od CSV až po XML, data zkoumat, filtrovat a kreslit podle nich pěkné grafy. Vyzkoušíme si prakticky v Pythonu. Předvedu své oblíbené nástroje, pojďte ostatním předvést ty své.

Matematické přednášky

Pravděpodobnost **PAST**
Michal Kodad, David Klement, Jirka Kalvoda
Jak pracovat s pravděpodobností matematicky. Ukážeme si pravděpodobnosti jevů, nezávislé jevy střední hodnotu, náhodné proměnné a další. Také si vše procvičíme na několika příkladech. Pravděpodobnost bývá mnohdy neintuitivní, proto poukážeme na časté nachytávky z reálného života. Pokud zbyde čas, tak si také ukážeme, jak se dá pravděpodobnost využít v informatice.

Fourierova transformace ** **FFT**

Martin „Medvěd“ Mareš

Jak rychle umíte násobit n-ciferná čísla? My to umíme lineárně. Hodí se k tomu chytrý trik pana Fouriera, který už dávno patří k matematické a fyzikální klasice. Ukážeme, co je Fourierova transformace zač, jak ji rychle spočítat a k čemu je dobrá: rychlé násobení polynomů i čísel, digitální zpracování zvuku a obrazu (spektrální analýza či třeba komprese).

Předpoklady: Základy komplexních čísel

Teorie (vesměs samoopravných) kódů *(“f y cn rd ths, y wll b gd cmptr prgrmmr!”)* **KODY**

David Klement, Martin „Medvěd“ Mareš, Jirka Kalvoda

Jak komunikovat po lince, která průměrně každý k-tý bit přenese špatně? K tomu se hodí teorie samoopravných kódů, která nás naučí: vzdálenost slov a jejich souvislost s detekcí a opravou chyb, paritní a lineární kódy, perfektní kódy, Reed-Solomonovy a vůbec polynomiální kódy a několik dolních odhadů nádavkem. A jak s teorií kódů souvisí třeba čeština?

Lineární programování jako blackbox * **LPBB**

Říša Hladík, Jirka Kalvoda

Lineární programování je ohromně užitečná optimalizační technika. V přednášce se nebudeme zabývat teorií, a raději si ukážeme co nejvíce praktických využití této techniky zejména pro návrh efektivních algoritmů. Od problémů, kde lineární programování číhá v přestrojení, až k efektivnímu řešení NP-těžkých problémů, pokud nám stačí jen řešení přibližné. Zaokrouhlování, randomizace, celočíselné programování a další triky.

Lineární a konvexní optimalizace * *(“Co mají společného toky v sítích, trénování neuronky a SAT?”)* **OPTOVR**

Říša Hladík

Přehledová přednáška o nejběžnějších metodách lineární a konvexní optimalizace. Ukážeme si několik algoritmů schopných vyřešit libovolný problém, který umíme zapsat pomocí lineárních a konvexních podmínek. Gradient descent, Newtonova metoda, Frank-Wolfe a jeho varianty. Lineární programování: simplexová metoda, elipsoidová metoda, metody vnitřního bodu. Jak dokázat, že nalezené řešení je optimální, aneb KKT podmínky. Spíš než o rigorózní přístup a důkazy se pokusíme vybudovat intuici.

Předpoklady: MA, základní tušení o lineární algebře

Lambda kalkulus * *(“Každý program v Pythonu jde zkrátit na jeden řádek.”)* **LAMBDA**

Dan Skýpala, Jirka Kalvoda

Pořídíme si ty nejkrotčejší funkce a potom z nich naprogramujeme vše. Jak výrazy s funkcemi zjednodušovat? A jak vybudovat z funkcí pravdivostní hodnoty a čísla? Funkce která vrací pevný bod funkce. A nakonec trocha Haskellu.

Komplexní a komplexnější čísla *(“$1 = \sqrt{1} = \sqrt{(-1)(-1)} = \sqrt{-1}\sqrt{-1} = i \cdot i = i^2 = -1$. Huh?”)* **CPLX**

Ján Plachý

Jak se nám matematika změní, když připustíme, že se záporná čísla také dají odmocňovat? Čísla imaginární a komplexní a jejich různé podoby. Součtové vzorce pro sin a cos dostaneme téměř zdarma. K čemu se hodí v matematice a k čemu ve fyzice. Proč se zastavit u dvou složek aneb kvaterniony, oktoniony a Cliffordovy algebry. Remember, life is complex.

Úvod do matematické analýzy * *(“$\lim_{8\to 9} \sqrt{8} = 3$”)* **MA**

Ondra Sladký, Honza Černý

Jak zjistit, jaký tvar má graf nějaké funkce? Jak najít její minimum? Jak spočítat délku spirály nebo objem sudu (třeba i čtyřrozměrného)? Jak spočítat $\sin x$ nebo třeba π? Na to všechno se hodí limity, derivace a integrály. Nejprve si o nich vybudujeme jednoduchou geometrickou představu, pak je nadefinujeme pořádně a naučíme se s nimi počítat.

Barevnost grafů * *(“Bílá, modrá, červená, co to pro graf znamená?”)* **BAGR**

Ondra Sladký, Říša Hladík, Jirka Kalvoda

V teorii grafů zaujímá významné místo problém barevnosti grafu, tedy přiřazení co nejmenší počtu barev vrcholům tak, aby se hranami dotýkaly pouze různobarevné vrcholy. Aplikace problému v informatice je nasnadě. Ukážeme si několik zajímavých teoretických výsledků. Obarvení některých druhů grafů, $L_{2,1}$ barevnost aneb problém vysílačů, vybíravost, kruhová barevnost a další.

Rovinné grafy *(“Kdo nakreslí pět souvislých států tak, aby každý sousedil s každým, má u mě čokoládu.”)* **ROG**

Ján Plachý, Jirka Kalvoda

Povídání o grafech, které jde nakreslit na papír bez křížení hran. Co všechno pro takové grafy platí a jak je poznáme, aniž bychom je museli kreslit. Existuje pouze 5 pravidelných mnohostěnů, a my se o tom pomocí teorie grafů přesvědčíme. Barvení rovinného grafu šesti a možná i méně barvami. Když zbyde čas, zkusíme grafy kreslit i na jiné plochy: kupříkladu Möbiovu pásku, pneumatiku nebo ušatou kouli.

Grafy bez algoritmů **GRAFY**

Říša Hladík, Jirka Kalvoda, Honza Černý

Teorie grafů trochu teoretičtěji. Různé druhy grafů a jejich vlastnosti. Stromy a lesy. Kreslení grafů jedním tahem. Princip sudosti a skóre grafu. Jaké speciální vlastnosti mají rovinné grafy a jak je lze obarvit šesti nebo možná i pěti barvami. Jak poznat, že dva grafy (ne)jsou isomorfní. Mosty, artikulace a ušaté lemma. Párování, střídavé cesty a Hallova věta.

Úvod do teorie čísel **NUT**

Říša Hladík, David Klement, Martin „Medvěd“ Mareš

Co a k čemu je teorie čísel. Počítání v kongruenci, Euklidův algoritmus a jeho použití. Konečná tělesa a Malá Fermatova věta. Prvočísla a Eratosthenovo síto. Čínská zbytková věta a její algoritmická verze. Jak si odvodit kritéria dělitelnosti.

Teorie čísel a RSA * *("$2^{67} - 1 = 193\,707\,721 \cdot 761\,838\,257\,287$")* **NUT2**
Martin „Medvěd" Mareš
Pokračování teorie čísel, které nás dovede až k RSA – asi nejpoužívanějšímu asymetrickému šifrovacímu algoritmu dnešní doby. Počítání modulo složené číslo a Eulerova věta. Jak RSA funguje, proč funguje a jestli bude ještě fungovat. Generování klíčů, faktorizace kontra testování prvočíselnosti. Časová složitost aritmetiky.

Kombinatorika *("Nemám rád faktoriály. Faktoriály nemám rád. Rád nemám faktoriály. . . ")* **KOMB**
Martin „Medvěd" Mareš, Jirka Kalvoda
Při navrhování algoritmů a počítání jejich složitosti narazíme na celou řádku zajímavých a ne úplně triviálních kombinatorických problémů, a tak se naučíme, jak na ně. Základní triky s faktoriály a kombinačními čísly, sčítání konečných a občas i nekonečných řad, rekurentní rovnice a princip inkluze a exkluze. Možná se také potkáme s Dlouhým, Širokým a poněkud zmatenou šatnářkou.

Teorie množin a matematika nekonečen * *("Kdo je nejvyšším z kardinálů?")* **TEMNO**
Říša Hladík, Martin „Medvěd" Mareš
Historie matematiky je dlážděna trampotami s nekonečnem. Začalo to roztomilým problémem s želvou pana Zénona a vedlo až k poněkud děsivým paradoxům 18. století. V moderní době jsme se proti tomu obrnili teorií množin, na níž je dnes takřka celá matematika postavena. Jak se taková teorie buduje a jak se pomocí ní popisují nekonečné objekty. Množiny a jejich velikosti. Cantorův diagonální trik. Ordinály a housť kardinálů. Potenciální kontra aktuální nekonečno. Jak si pořídit přirozená čísla a jak ta reálná. Potíže s axiomem výběru.

Základy algebry * **ALGBR**
Říša Hladík, Martin „Medvěd" Mareš
Jak matematici dokáží vzít „obecné" a ještě více jej zobecnit. Ukážeme si, jak zkoumat matematické operace, aniž bychom řešili, jestli se bavíme o sčítání, násobení, nebo skládání zobrazení. Pár magických slov pro představu: monoidy, grupy, okruhy, obory integrity, tělesa, vektorové prostory, polynomy, částečná uspořádání, booleovy algebry, filtry, ideály. Také si povíme, na co se takové věci dají použít – a jak to celé souvisí s prvočísly, jak se šifrováním a jak s osovou souměrností.

Teorie her *("Někdy hloupé chování každého je rozumnou reakcí na hloupé chování někoho jiného")* **GAMTH**
Říša Hladík
Ukážeme si, jak jednoduché principy z matematické disciplíny teorie her vysvětlují mnoho na první pohled zvláštních jevů v lidské společnosti i přírodě – poškozování životního prostředí, monopol vědeckých časopisů, jaderné zbrojení, úspěch Facebooku, předražené zboží a další. Povíme si o známém vězňově dilematu a problému obecní pastviny. Seznámíme se s pojmem Nashova ekvilibria – aneb proč někdy všichni hráči dělají rozhodnutí, které vede na pro ně nepříznivý výsledek, ale nikdo není motivován své rozhodnutí změnit. Ukážeme si, že se občas paradoxně vyplatí omezit si vlastní možnost volby. Další aplikace: aukce, vyjednávání, volební systémy, . . .

Logika aneb jak se staví matematika *("Následující věta není pravdivá. Předchozí věta je pravdivá.")* **LOGI**
Martin „Medvěd" Mareš, Honza Černý
Pokud budeme v životě věřit všemu, co je „přeci zřejmé", dostaneme se brzy do potíží a v matematice to platí dvojnásob. Proto své teorie musíme stavět pečlivě. Na to si filosofové a matematici pořídili logiku. Ukážeme, jak funguje výroková a predikátová logika, co je to výrok, axiom a důkaz. Vybudujeme pár jednoduchých teorií a podíváme se, co dovedou, a co už ne. Důkazy za nás ověří počítač, aspoň když mu trochu pomůžeme. Nadšení trochu ochladí Gödelova věta: ať děláme, co děláme, vždy zbude nějaké nerozhodnutelné tvrzení. Pomůže přidávat axiomy? Asi ne, ale za odměnu získáme mnoho různých matematik. A dá-li bůh, stihneme dokázat jeho existenci i neexistenci ☺.

Catalanova a Fibonacciho čísla * *("1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, ?")* **CAT**
Martin „Medvěd" Mareš
Kolik existuje binárních stromů? Kolika způsoby jde uzávorkovat výraz? A kolika způsoby projít čtvercovou mřížku, aniž bychom překročili úhlopříčku? Kam oko pohlédne, všude se skrývají Catalanova čísla. Kromě případů, kdy za ně zaskakují čísla Fibonacciho. Povídání o dvou zajímavých posloupnostech a jejich početném příbuzenstvu. Dlouhá cesta od hezkého vzorečku k rychlému algoritmu.

Křivky v počítačové grafice * *("Jak se měří elegance křivky?")* **BEZI**
Martin „Medvěd" Mareš
Jak se na počítači kreslí křivky, které „vypadají hezky", třeba tvar karoserie auta nebo tvar písmenka? Kružnice a jiné kuželosečky se k tomu moc nehodí, tak se poohlédneme po obecnějších křivkách. Základy matematiky okolo Bernsteinových polynomů, Bézierových křivek a spline funkcí. Práce s křivkami pomocí rekurzivního rozkladu a de Casteljauova algoritmu. Matematické modelování estetiky.
Předpoklady: Pro část přednášky se hodí vědět, co je derivace, a nebát se ji použít.

Hausdorffův zvěřinec ** *("Jaký objem má π-rozměrná koule?")* **HAUS**
Martin „Medvěd" Mareš
Možná vás už také zarazilo, že některé fraktály nejsou ani dvourozměrné, ani třírozměrné, ale něco mezi tím. Pojďme se podívat, co to znamená. Cestou potkáme různé zajímavé partie matematiky (jako třeba metrické prostory a teorii míry) a různá podivuhodná zvířátka: Cantorovo diskontinuum, von Kochovu vločku a Hilbertovu křivku.

Ostatní přednášky

Lingvištika *("Přísudek je v této větě podmět.")* **LING**

Martin „Medvěd" Mareš

Převážně nevážné a mírně nepřed-vídatelné po-vídání o jazyku i jazyce. Základní jazykové rodiny a jejich podobnosti i odlišnosti. Co má společného čínština s angličtinou a co nikoliv. Proč jeden jazyk potřebuje 15 pádů, zatímco jiný se bez nich obejde úplně. Jak se jazyky vyvíjejí a jak se navzájem ovlivňují. Kde se berou jazyková pravidla. Kde se vzalo písmo a proč se mluvený a psaný jazyk tolik liší. Jak se na jazyk dívá matematik a jak se na matematiku dívají lingvisté.

Fonetika *("Pojďte, zachrochtáme si spolu!")* **FON**

Martin „Medvěd" Mareš

Malá inventura zvuků, které lidé dovedou vytvářet, a jejich použití v komunikaci. Různé způsoby vytváření a modulace zvuku. Kolik různých B dokážete říci? Fonetické kontrasty a co si z nich různé jazyky vybraly. Rázy, polosamohlásky a jiní obyvatelé polosvěta. Přízvuk kontra délka. Asimilace, přehlasování a další „principy líné huby." Vše prakticky procvičíme.

Orientace **ORI**

Martin „Medvěd" Mareš

Jak ze neztratit v terénu a jak se neztratit na moři. Vývoj umění navigace. K čemu je důležité slunce a hvězdy, ale proč mořeplavcům nestačí, alespoň dokud neobjevíme hodinky. Použití mapy, busoly a GPSky. Orientace bez pomůcek a použití Ariadniny nitě. Bleskový úvod do sférické astronomie a časomíry čili jak (ne)postavit sluneční a třeba i měsíční hodiny. Jak reprezentovat mapu v počítači a jak raději ne. Jak zapisovat polohu místa na Zemi (přestože Země má tvar podivně nakousnuté hrušky) a kolika způsoby to jde. Různé druhy map a jejich (z)kreslení. Jak se neztratit v kartografii. Praktické cvičení v terénu.

Jak si pořídit vlastní jazyk *("Minao remabi malelio koribeto.")* **CONLANG**

Martin „Medvěd" Mareš

Hodí se vám, aby postavy ve vaší hře nebo povídce mluvily neznámým jazykem? Tak si vymyslete vlastní! Ale jak na to? Jak se vytváří různé vrstvy jazyka: slovní zásoba, morfologie, gramatika, frazeologie, ale i písmo a výslovnost. Proč si pořídit imaginární uživatele a imaginární historii. Jak najít správnou míru nepravidelnosti. Čím se můžeme inspirovat z existujících jazyků a čím raději nechceme.

Čárové kódy *("Jak naučit počítače číst láhve od Coly.")* **BAR**

Martin „Medvěd" Mareš

Čárové kódy dnes potkáváme na každém kroku, ale jak doopravdy fungují? Prozkoumáme klasické jednorozměrné kódy (UPC, EAN, Code39, Code128), jakož i novější dvojrozměrné (QR, Aztec, DataMatrix). Kódovací a dekódovací algoritmy plus trocha matematiky okolo zabezpečení proti chybám. Další počítačem čitelné značky: RFID, bíle křížky na asfaltu, ...

Půlnoční přednášky

Aneb přednášky přednášené (nejen) o půlnoci na různá zajímavá témata nejen o informatice. Pokud nějaká z nich nebude oficiálně vypsaná, je možné si konkrétního organizátora ve volné chvíli chytit a přesvědčit ho k přednášení.

Organizování a práce v týmu *("Ten dělá to a ten zas tohle aneb co obnáší organizátorem býti.")* **ORG**
Jirka Kalvoda
Volné povídání o tom, co se všechno skrývá za organizováním různých seminářů a podobných akcí, primárně pak KSPčka. Jaká práce, jaké radosti a jaké starosti s sebou organizování nese, co se přitom člověk může naučit a také pár cenných rad do života. Jak se z toho nezbláznit a pár bláznivých příhod k tomu.

Základy první pomoci *("Jak někomu zachránit život a jak málo k tomu stačí.")* **ZDRAV**
Říša Hladík
Pobavíme se o základech první pomoci. Jak správně vyhodnotit situaci a kdy je potřeba volat pomoc? Jak se postarat o člověka v bezvědomí, jak kontrolovat životní funkce a jak člověka stabilizovat do příjezdu pomoci? Ukážeme si, jak málo stačí k záchraně života a naučíme se nebát se první pomoci. A také, že naše bezpečí je v každé situaci na prvním místě.

Klávesové zkratky *("`Ctrl+Shift+Alt+Super+J`")* **KEYB**
David Klement, Jan Černohorský, Jirka Kalvoda
Jakmile se rozhodnete ovládat počítač klávesnicí, brzy zjistíte, že anglická abeceda má příliš málo písmenek pro všemožné klávesové zkratky. Nemluvě o tom, jak si všechny zkratky zapamatovat. Přednáška až diskuze o různých přístupech, které výše zmíněné problémy řeší.

Čaj *("Jak vypadá odvar z nezralých pražců?")* **TEA**
Martin „Medvěd" Mareš, Honza Černý
Pojďme usednout k šálku lahodného čaje a povídat si o tom, co se v něm skrývá. Kde se čaj vzal, kde se pěstuje, jak se zpracovává a jak ho připravovat. Trocha čajového zeměpisu, dějepisu i čajové chemie a čajové kultury. Též o všelijakých substancích čaji podobných.

Nová náboženská hnutí *("Vše, co potřebujete vědět, abyste si mohli založit sektu.")* **CULT**
Honza Černý
Od poloviny minulého století přibylo mnoho nových náboženství a duchovních hnutí. Povíme si o jejich historii a dopadu na společnost. Povíme si i o psycholigii schovanou za sektami a jak vlastně interně fungují. Povíme si i o charismatizaci a jak ji poznat.

Hnutí nového věku *("Jak vyrobit boha na míru.")* **NEWAGE**
Honza Černý
Žijeme v době, kdy můžeme být svědkem velkých změn, a to dokonce i ve vnímání duchovna a životních hodnot. Řekneme si o historickém vývoji náboženského směru "new age" a jeho aktuální podobě. Cílem přednášky je pochopit vývoj hodnotových systémů a jak ovlivňuje jednotlivce. Přednáška z velké části vychází z pozorování Karla Gustava Junga.

Střípky hudební teorie *("CΔ+79♭11♯/Gadd2/Dsus4/Amøn/Gu5")* **MUSIC**
Říša Hladík
Volnější přednáška na téma hudba, která se po zadefinování základních axiomů (půltón, doba, stupnice, akord) volně vydá tam, kam nás vítr zavane. Proč zní ten samý tón jinak na různé nástroje, když je to vždycky ta samá frekvence? A je to s těmi 12 půltóny a třemi stupnicemi tak jednoduché, jak říkají v ZUŠce? Jaké všechny akordy existují a proč jich většinu v populární hudbě nepotkáte? Jak akordy fungují spolu a proč ten samý akord může znít stokrát jinak? Jak to melodie celé drží pohromadě? A proč je vůbec populární hudba tak populární? Bude-li čas, můžeme si zkusit vytvořit nějakou vlastní "tucku", zanalyzovat písničku, se kterou přijdete, nebo si o tom všem zafilozofovat.

Orientační běh s migrujícími kontrolami *("Ten, kdo tohle vymyslel, musel být matfyzák.")* **SMIK**
David Klement
Jednou ročně se sejde pár stovek šílenců, aby běhali po lese, snažili se neztratit, a během toho všeho řešili problém obchodního cestujícího dotažený do extrému. Tak nějak vypadá SMIK. Představíme si jeho princip, prohlédneme mapy z proběhlých závodů a zkusíme na nich najít nejlepší cestu.

Lockpicking *("Jak si odemknout, když si náhodou my (nebo soused) zapomeneme klíč :-)")* **PICK**
Honza Černý
Jak fungují dnešní zámky, co jsou to stavítka a jak vlastně fungují klíče. A jak se pomocí jednoduchých nástrojů dají využít výrobní nedokonalosti zámků k jejich odemčení. Použití planžet, napínáků, praktické ukázky odemykání, nastínění technik bumpingu a dalších postupů, jak se dostat přes zamčené dveře.

Jazyková Zoo *("Na co GO TO? Máme COME FROM.")* **JZOO**
Martin „Medvěd" Mareš
Obecná teorie programovacích jazyků má asi tolik půvabu, jako biologická systematika. Tak se raději pojďme podívat do zoo: poznejme jazyky klasické, experimentální i dočista absurdní. Ada, Céčko a Python (tři pohledy na fungování typů). Praděde-ček všech funkcionálních jazyků LISP (program a data jsou totéž). APL (algebraické inspirace, nebo též průvan ve skladišti písmenek). Forth (zásobníkový předchůdce Postscriptu, ale i javovského virtuálního stroje). Lingua::Romana::Perligata (programovací jazyk, který skloňuje a časuje). Shakespeare, Intercal, Oook! a jiné komedie. Samorozšiřitelné a hybridní jazyky.

Abecední seznam přednášek

Základní přednášky

Půlnoční přednášky