

PODZIMNÍ SOUSTŘEDĚNÍ KSP 2016 – 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 fórum), 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**

Zuzka Drázdová, Karry Burešová, Jenda Hadrava

Pravděpodobně dvoudílná přednáška pro ty, kdo potřebují dohnat základní znalosti nutné pro ostatní přednášky. Zdefinujeme 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**

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.

Prohledávání do hloubky **DFS**

Trochu hlubší pohled na prohledávání do hloubky. Jeho (často dost nečekané) aplikace v dalších algoritmech, jako je třeba hledání mostů, topologické třídění, rozklad na komponenty silné souvislosti či kreslení grafu jedním tahem.

Toky v sítích (“*Když je v grafu povodeň, těsní?*”) **TOKY**

Jirka Setnička, Petra Pelikánová, Jenda Hadrava

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)

Datové struktury pro začátečníky (“*Pole oraná a neoraná, stromy ovocné a okrasné.*”) **DS1**

Jak si ukládat data natolik šikovně, abychom je nejen neztratili, ale také našli dříve, než si pro nás přijde Smrt. 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**

Jirka Setnička, Karel Tesař, Martin Mareš, Jenda Hadrava

Důmyslnější varianty vyhledávacích stromů: splay stromy, BB - α stromy, vícerozměrné stromy. Chytřejší haldy: binomiální, Fibonacciho, 2-3. Amortizovaná analýza složitosti. Též několik přátelských randomizovaných datových struktur: skip listy a treapy.

Intervalové stromy * (“*Já bych ty intervaly nejradši. . . dal do stromu!*”) **ITREE**

Jirka Setnička, Karry Burešová, (Jenda Hadrava)

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**

Karry Burešová, Karel Tesař, Jenda Hadrava

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?»*)

REGEX

Jirka Setnička, Karry Burešová, Petra Pelikánová, Jenda Hadrava

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.

Parsing čili analýza textu (“*1+2*4 = 12*”)

PARSE

Karry Burešová, Martin Mareš

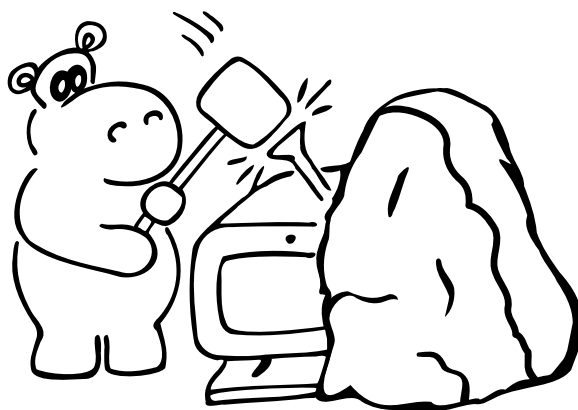
Často potřebujeme načíst nějaký složitý textový vstup: matematický výraz, webovou stránku v HTML, zdroják programu, . . . Ukážeme si, jak texty analyzovat (neboli parsovat), aniž bychom v nich zabloudili: rozdělení na lexikální a syntaktickou vrstvu, železničářský algoritmus na parsování výrazů, popis syntaxe pomocí regulárních výrazů a gramatik.

Geometrie a počítače (“*Nerušte mé kruhy! (ani jiné kvadriky)*”)

GEOM

Jirka Setnička, Karel Tesař, Martin Mareš, Jenda Hadrava

Základní algoritmy pro řešení geometrických úloh – konvexní obal, dva nejbližší body v rovině, výpočet obsahu nekonvexního mnohoúhelníka, lokalizace bodu, scanline algoritmus a jeho použití, Voroného diagramy a souvislost s persistentními datovými strukturami.



Programovací jazyky a nástroje

Programování v jazyce C

C

Kuba Maroušek, Jenda Hadrava

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 knihovnamí jazyka C.

Objektově orientované programování nejen v C++ (“*Object-oriented system. If we change it, users object.*”)

OOP

Štěpán Hojdar, Kuba Maroušek

Objektově orientované programování přináší jiný náhled na návrh řešení problémů. Vysvětlíme, jak se liší objektové a procedurální programování. Co je to objekt a co třída. Základní vlastnosti objektů (dědičnost, zabalení, polymorfismus). Co je to metoda, překrývání metod, virtuální metody (pozdní vazba) a čistě virtuální (abstraktní) metody. Syntaxe a odlišnosti ve třech nejvíce používaných jazycích – C++, C# a Java.

Předpoklady: Znalosti procedurálního programování, například v Pascalu, v Pythonu nebo v C.

Programování v jazyce C# (“*Co se stane, když strčíme Cěčko za mříž?*”)

CIS

Katka Zákravská

C# je moderní objektově orientovaný jazyk, jehož tvůrci se inspirovali přednostmi a úskalími ostatních programovacích jazyků, zejména Javy. Je jednoduchý a crossplatformní (tedy snadno v něm vytvoříte i okýnka, která nepoběží jen na okýnkách). Naučíme se základy a možná si i napíšeme jednoduchý prográmek.

Předpoklady: Tušit něco málo o objektovém programování.

Python (“*print "Ffff".decode("rot13")*”)

PYTH

Martin Šerý, Filip Štědranský

Jak programovat v Pythonu a jak v něm „nepsat Cěko“. Syntaxe, datové typy, funkce, třídy, . . . Na co si dát pozor, v čem se Python liší od ostatních jazyků a proč je mezi nimi tak oblíbený.

Perl (“*Jak Pejsek a Kočička vymýšleli programovací jazyk*”)

PERL

Karry Burešová, Jirka Setnička, Kuba Maroušek

Jednoho dne se Larry Wall rozhodl, že nasype do jednoho velkého kotle spousty programovacích jazyků a unixových utilit, za stálého míchání povaří, posléze přecedí, přikoření a implementuje. Tak vznikl Perl, jazyk původně určený hlavně na zpracování textu, ovšem jak se ukázalo, též šikovný na spoustu dalších věcí. Asociativní pole, libovolně složitě datové struktury za pomoci referencí, balíčky a objekty zdarma a hlavně regulární výrazy zde a všude. Zkrátka jazyk, který lze jedinečně milovat nebo nenávidět, nic mezi tím. Co se Perl 5 přiučil od Perlu 6.

- Procesy a vlákna *** (*“Koupil jsem dalších 15 procesorů, proč je to stále stejně pomalé?”*) **THREAD**
Filip Štědrónský, Martin Mareš, Jenda Hadrava
 Jak vypadá víceprocesorové či vícejádrové PCčko a co to znamená pro programátora. Procesy, vlákna a úskalí komunikace mezi nimi. Jak se snese n kohoutů na jednom smetišti? Synchronizační primitiva: mutexy, semaforey, podmínkové proměnné. Spinlocky, deadlocky a livelocky. Jde to i bez synchronizace: atomické operace, transakční paměť. Které jazyky nám pomáhají a které spíš škodí. Kdy je lepší vlákna použít, a kdy ne.
Předpoklady: Trochu představy o hardwaru
- Logické programování** (*“Detektivem za 90 minut.”*) **LOGP**
Jenda Hadrava
 Proč psát dlouhé a složité programy, když stačí dostatečně přesně popsat situaci a pak se prostě zeptat? Toť princip logického programování, který si ukážeme na Prologu.
- Haskell** (*“V téhle proměnné je uložen okolní svět.”*) **HASK**
Jenda Hadrava
 Základní kurz Haskellu – moderního funkcionálního jazyka. Ukážeme si syntaxi, vysvětlíme typovou kontrolu a typový systém. Přičichneme k třídám, zrušíme výjimky a zavedeme zcela bezpečná vlákna. Řekneme si, proč v Haskellu nejde komunikovat s okolním světem a proč nám pomůže si okolní svět uložit do proměnné. A že vlastně v Haskellu žádné proměnné nejsou, jen visáčky na datech.
Předpoklady: Sklony k algebraickému chápání vesmíru, odvahu tváří v tvář své vlastní tváři a rekurzi.
- Dynamický web a PHP** (*“Pepičku, napíšeš mi é-šopík?”*) **PHP**
Kuba Maroušek, Karry Burešová
 Základy praktické tvorby dynamického webu. Úvod do jazyka PHP a Javascriptu, čtení dat z odeslaných formulářů, přesměrování, databáze, generování obrázků a další.
- Jazyk SQL** (*“SELECT something FROM knowledge LIMIT 90min”*) **SQL**
Kuba Maroušek, Karry Burešová
 Dotazovací jazyk SQL a jeho aplikace, čili jak se domluvit s relační databází a zeptat se rovnou na to, co chci vědět. Definice tabulek a indexů. Dotazy a jejich skládání a vnořování. Pohledy, funkce a trigger. Rozdíly mezi dialekty SQL.

Hardware a operační systémy

- Principy počítačů** (*“A opravdu uvnitř počítače běhají malí trpaslíci?”*) **HW**
Jirka Setnička, Martin Mareš, Jenda Hadrava, Filip Štědrónský
 Vydáme se do země skřítků, kteří pohánějí počítače. Počítačové architektury od hodiniek 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.
- Od zdrojáku k programu** (*“Před spuštěním program přeložte. Stačí třikrát podělně?”*) **KOMP**
Filip Štědrónský, Martin Mareš
 Mezi programem v Céčku, který jste právě dopsali, a tranzistory uvnitř vašeho procesoru leží obrovské území obývané překladači, linkery, knihovníky, operačními systémy, loadery a jinými bájnými bytostmi. Pojdme zjistit, co jsou zač a co všechno s programem provádějí. Co udělá kompilátor za nás a co musíme naopak udělat my za něj.
- UNIX** (*“UNIX gives you enough rope to hang yourself.”*) **UNIX**
Jirka Setnička, Karry Burešová, Filip Štědrónský, Jenda Hadrava
 Unixové operační systémy (zejména Linux) dobývají svět. Jak fungují uvnitř a jaké nabízejí výhody? Unixová filosofie a historie. Proč je systém složený ze spousty malých a jednoduchých kousků stabilnější a bezpečnější? Proč ovládání prostřednictvím textových příkazů je často efektivnější než klikátka? Jaké to je mít svůj systém pod kontrolou a „vidět mu pod ruce“? V čem spočívá moc textových souborů?
- Skriptování v shellu** (*“man 1 woman ... man 2 woman ... man group”*) **SHELL**
Filip Štědrónský, Jenda Hadrava, Karry Burešová
 Praktičtěji zaměřená přednáška než UNIX, zabývající se hlavně tím, jak efektivně používat příkazovou řádku. Ukážeme si na spoustě příkladů, jak nám může automatizace všedních činností ulehčit život a jak silné nástroje pro ni UNIXový shell (který navzdory svému názvu existuje i pro Windows) svou jednoduchostí a flexibilitou poskytuje. Budeme spojovat spoustu jednoduchých příkazů do mocných celků a s nimi plnit i na první pohled komplexní úlohy, jako třeba automatické stahování a parsování věcí z webu. Některé činnosti vyžadují lidskou nápaditost a vhléd. Ty ostatní bychom měli přenechat strojům.
- Programování v Linuxu** **PLX**
Filip Štědrónský, Martin Mareš, Jenda Hadrava
 Jak si program pod Linuxem povídá s operačním systémem, když chce otevřít soubor, přečíst soubor, půjčit trochu paměti a jiná šprťouchlata. Předvedeme si, jaká existují v Linuxu systémová volání. Naučíme se namapovat si soubor rovnou do paměti, posílat a odchytávat signály, uspávat a probouzet proces, plodit děti a další. Pokud zbyde čas, můžeme si napsat démona a klienta a povídat si po síti.
Předpoklady: Schopnost přečíst a napsat jednoduchý program v C.

Sítě a bezpečnost

Sítě a Internet (“*Sítě nejen na ryby.*”)

NET

Martin Mareš, Kuba Maroušek, Filip Štědranský

Jak funguje Internet a počítačové sítě vůbec: od elektronů v drátech (fotonů v optických kabelech nebo elektromagnetických vln) přes pakety a jejich routing 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 Mareš, Filip Štědranský

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

Kryptografie (“*Gbg arav zbp gnwan mcenin.*”)

CRYPT

Filip Štědranský, Martin Mareš, Karry Burešová

Kryptografie čili tajuplná nauka o šifrách, jejich konstrukci a hlavně o jejich luštění. Přísně tajné. Šifrovací systémy jako lego: základními kostičkami nám budou symetrické a asymetrické šifry a jednosměrné funkce, 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.

Teoretická informatika

Složitější složitost *

SLOZ2

Jirka Setnička, Karry Burešová, Jenda Hadrava

Trochu hlouběji o složitosti. Přesná definice výpočetního modelu a velikosti vstupu. Složitost v nejlepším, nejhorším a průměrném případě; amortizovaná analýza. Jak dokázat, že úlohu nejde řešit rychleji, aneb dolní odhady. Porovnávání problémů pomocí redukcí, problémy NP-úplné a ještě těžší.

Předpoklady: SLOZ

Umělá inteligence *

AI

Karel Tesař

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.

Evoluční algoritmy * (“*Já to dělat nebudu, ať to za mě udělají mravenci!*”)

EVA

Jirka Setnička, Karry Burešová, Karel Tesař

Evoluční algoritmy se inspiřují strukturami chování v přírodě a na jejich základě pak (optimalizačně) hledají řešení těžkých problémů. Na přednášce určitě zazní genetický algoritmus, zmíníme jeho algoritmy a když zbyde čas tak si obecněji popovídáme o algoritmech pohybujících se ve velkých prostorech řešení.

Strojové učení * (“*Umí počítače přemýšlet?*”)

MACHINE

Karel Tesař, Karry Burešová

Úlohy, kterými se zabývá strojové učení. Co je to učení s učitelem a bez učitele? Ukážeme si základní postupy, jak počítač můžeme naučit rozpoznávat věci, které sami rozpoznat neumíme. Umí Facebook nebo Google poznat, co máme rádi, jakou máme náladu, nebo jestli jsme těhotní?

Modely počítačů (“*Nač Pentium? Máme Turingovy stroje!*”)

MODEL

Jirka Setnička, Martin Mareš, Jenda Hadrava

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. 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.

Jazyky, gramatiky a automaty *

AUTO

Martin Šerý, Karry Burešová, Jenda Hadrava

O jazycích přirozených, počítačových 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?

Matematické přednášky

Grafy bez algoritmů

GRAFY

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 (*“Po malém fermetu mívám čínský zbytkáč.”*)

NUT

Martin 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.

Kombinatorika (*“Nemám rád faktoriály. Faktoriály nemám rád. Rád nemám faktoriály. . .”*)

KOMB

Petra Pelikánová

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.

Intervalové počítání (*“Prima, sekunda, tercié, . . . nekonečno.”*)

INT

Petra Pelikánová

Co s čísly, která si počítač nezapamatuje přesně? Vezmeme si na pomoc intervaly a zabalíme je do nich. Naučíme se s intervaly počítat a nakrmíme nimi i funkce. Řekneme si, kde se dají využít v praxi, ale i v teoretické informatice.

Rozšiřující přednášky

Mezi rozšiřujícími přednáškami se dají nalézt různé specifitější obory a zájmy, jakožto i těžší přednášky navazující na předchozí díly ze základních přednášek. Mezi nabízenými přednáškami si tak můžete vybrat obor svého zájmu a tomu se dále věnovat.

Algoritmy a datové struktury

Datové struktury pro šílence ** (*“A tohle fakt jde naprogramovat? Tomu nevěřím!”*)

DS3

Karel Tesař, Martin Mareš

Jak postavit datovou strukturu, která si pamatuje svou vlastní historii? Jak z libovolné statické datové struktury udělat dynamickou? Lze navrhnout vyhledávací strom se složitostí $O(\log \log n)$? A jak pro danou sadu prvků postavit strukturu, které v nich vyhledává v čase $O(1)$? Tyto a ještě další triky budou obsaženy v této přednášce.

Stromové algoritmy (*“Půjdem na to od lesa”*)

TREES

Martin Mareš

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, rozklad na lehké a těžké hrany, Fredericksonův rozklad a ST-stromy.

Suffixové stromy ** (*“Jak obrátit řetězec naruby?”*)

SUFF

Filip Štědranský, Martin Mareš, Karry Burešová

Suffixový strom je zajímavá datová struktura, pomocí níž jde vyřešit většinu řetězcových problémů v lineárním čase. Podíváme se, jak suffixový strom vypadá, k čemu se hodí a jak ho sestrojít. Též prozkoumáme několik příbuzných zvířátek, jako třeba suffixové pole a suffixový automat.

Nejkratší a jiné cesty * (*“Všechny cesty vedou do Horní Dolní, jen některé přes Řím.”*)

CESTY

Jirka Setnička, Karry Burešová, Filip Štědranský, Jenda Hadrava

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.

Magické algoritmy * (*“Pokročilá magie není rozlišitelná od technologie.”*)

MAGIC

Martin Mareš, Jenda Hadrava

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 kejklřství. Hledání největší díry.

Programovací jazyky

C for wizards * (*“1[x]+++++x[1]”*)

CWIZ

Martin Mareš

Ponořme se do hlubin Cčka, snad až na samé dno. Typový systém: elementární typy, typové výrazy, automatické konverze a rozpad typů (pole vs. ukazatel). Pořadí vyhodnocování kontra pořadí side-efektů (priority, synchronizační body a volatily). Triky s preprocesorem. Návěští a příkaz switch. Všelijaké zrady (velikosti typů, zarovnání, $(a + b) + c \neq a + (b + c)$, ...). Dialekty Cčka od K&R až po novou normu C11 a různá nestandardní rozšíření jazyka. Proč jsou objekty potřebnější v myslí programátorově než v jazyce a proč je C lepší než C++ ☺

Předpoklady: Povšechná znalost jazyka C.

Černá magie v C++ * (*“Je dobré znát, co umí atomová bomba (a její datový typ), abychom ji nechtěli použít.”*)

CPP

Filip Štědranský

Pokročilejší prvky C++ (šablony, přetěžování funkcí a operátorů, preprocesorové hacky, ...) a možnosti jejich (po|zne)užití. Jak s nimi vytvořit věci magické (smart pointers, lambda funkce), ale i na první pohled docela obyčejné, které si bohužel jinak pořídit nejde. Nahlédneme do vnitřností slavné knihovny Boost. Proč je třeba preprocesorem vygenerovat 512 šablonovaných typů jen, abychom mohli rozumně předávat pointery na metody? Jaká akrobacie je nutná pro obyčejnou statickou inicializaci seznamu? A mnohá další překvapení. Přednáška vás naučí psát v C++ složité věci a přesvědčí, že v něm nechcete psát ani jednoduché.

Předpoklady: základní znalost C++, staticky alokovaný kyblík.

Perl 6 (*“Slečno, mohu vám ukázat svou sbírku operátorů?”*)

PERL6

Filip Štědranský, Martin Mareš

Je to Perl, a přitom to Perl není. Co je to? Aneb jak to dopadne, když se pokusíme navrhnout programovací jazyk budoucnosti a inspirovat se přitom filosofií Perlu. Typový systém, pokud zrovna chcete. Objekty, třídy a metatřídy. Periodická soustava (meta)operátorů. Definování jazyka v sobě samém. A co se to stalo s regulárními výrazy? Jak vypadají implementace P6 a kdy je prozatím lepší programovat na papíře. Praktické cvičení ve stavbě vzdušných zámků a bydlení v nich.

Filip Štědranský

Povídání o méně zmiňovaných částech Pythonu. Dekorátory, metaklasy, generátory, funkcionální styl programování v Pythonu. Jak napsat quicksort jako lambda funkci. Představení zajímavých modulů nejen ze standardní knihovny. Další témata dle přání účastníků: paralelní programování (asynco, multiprocessing), propojení Pythonu s C, ...

Předpoklady: PYTH

Programování v assembleru

PASM

Martin Mareš

Jak programovat procesor přímo, aniž by vám do toho mluvily překladače, linkery a podobná verbež. Začneme obecně, ale soustředíme se hlavně na procesory rodiny x86. 32-bitová a 64-bitová instrukční sada, FPU a panoptikum vektorových instrukcí. Rozdíly mezi intelovskou a AT&T syntaxí. Jak spojit assembler s vyššími programovacími jazyky. Optimalizace kódu. Stručný úvod do systémových architektur IA32 a AMD64.

Jazyk Go

GOLANG

Jirka Setnička

Go je moderní kompilovaný jazyk (vyvinutý původně v Google), který se pokouší být takovým Cčkem na steroidech. Umí být podobně rychlý, zaručuje větší typovou bezpečnost, ale má i prvky dynamicky typovaných jazyků. Jeho velkou silou je velmi snadné provázání na existující kód v C/C++, systém balíčků distribuovaných převážně přes Github, funkce vracející libovolný počet hodnot nebo třeba vestavěná podpora Unicode a vestavěná hashovací tabulka. Také se silně dbá na coding-style pro zajištění snadné čitelnosti programů.

Programovací nástroje a techniky**Git a jiné systémy pro správu verzí** (“U svatýho tučňáka, kdo sem napsal tohle? Ono to tvrdí, že JÁ?!”)

GIT

Jirka Setnička, Karry Burešová, Martin Mareš, Jenda Hadrava

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. Kouzelnické triky: hledáme bugy pŕlením historie, přepisujeme dějiny. Jak se liší správa zdrojáků v projektech o jednom, deseti a tisíci programátorech. Udržujeme patche k cizímu programu aneb quilt a StGit.

Make (“make love ... don't know how to make love”)

MAKE

Jirka Setnička, (Jenda Hadrava)

Hodil by se otrok, který by překládal jednotlivé soubory. Základní syntaxe takového otroka, jak napsat jednoduchý Makefile, který řeší překlad Céčkového programu, automatické řešení závislostí. Jak to udělat, aby výsledek neměl několik tisíc řádek. Proč by se hodilo, aby tu bylo něco lepšího. A proč automake, cmake a qmake nepomáhají.

Gdb a jiné ladicí nástroje * (“Jak se ladí kytara, jak křišťálová koule a jak program (řazeno dle obtížnosti)”)

GDB

Filip Štědranský, Martin Mareš, (Jenda Hadrava)

Kdo píše programy, které vždy hned fungují, ať se přihlásí. A kdo ne, ať se přihlásí na tuto přednášku. Ladit programy vás nenaučíme, ale ukážeme několik nástrojů, které mohou pomoci z nejhoršího. Mezi nimi třeba gdb – řádkový debugger (odšívovač), strace, valgrind a perf. Kdy je chceme použít, a kdy se naopak víc hodí printf a assert.

Textový editor Vim (“Víš, jaký je nejlepší textový editor? Vim.”)

VIM

Karry Burešová, Filip Štědranský, Martin Mareš, (Jenda Hadrava)

Odložme na chvíli své myši 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, práce s regulárními výrazy, makra, kouzla. Vimovité ovládání jiných programů, třeba webového prohlížeče.

Jak se nestat vepřem (“/* You are not expected to understand this */”)

STYLE

Martin Mareš, Martin Šerý, Kuba Maroušek

Tvrdí se, že čistý 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 a jak napsat užitečný komentář nebo dokumentaci. A kdy se vyplatí se na všechna tato pravidla vybodnout.

Kompresce dat (“Jnm idln kpln j nstlčtln.”)

PRESS

Jirka Setnička, Filip Štědranský, Martin Mareš

Přehled základních kompresních algoritmů: triviální algoritmy (RLE), statistické metody (Huffmanovo a aritmetické kódování), slovníková komprese (LZ77, LZ78, LZW), Burrowsova-Wheelerova transformace (BZIP). Pokud zbude čas, tak i něco o ztrátové kompresi obrázků a zvuku (prediktory, wavelets, JPEG, MPEG, fraktály).

Hardware a operační systémy**Filesystemy** (“Opravdu je FAT tabulka tlustá?”)

FS

Filip Štědranský, Martin Mareš, (Jenda Hadrava)

Povídání o tom, co leží mezi nulami a jedničkami na plotnách disku a přátelskou adresářovou strukturou našeho OS. Jak funguje FAT a jeho varianty (VFAT, FAT32). Tradiční Linuxové filesystemy od EXT2 k EXT4. Nadějný nový BtrFS, který je možná za pár let nahradí. Co se hodí na SSD.

Jenda Hadrava

Jak fungují digitální elektronické obvody, ze kterých jsou postavené (nejen) počítače. Nuly a jedničky jako napěťové úrovně; kombinační obvody (transistory, hradla, multiplexery), sekvenční obvody (klopné obvody, registry, čítače) a asynchronní obvody. Troška matematiky okolo aneb logické formulky a De Morganovy zákony; proč stačí jenom jeden typ hradel. Třístavová hradla a sběrnice... zde plynule přecházíme v HW.

Cache-oblivious algoritmy (*“Kešuješ, kešuje, kešujeme”*)

CACHE

Jirka Setnička, Martin Mareš, (Jenda Hadrava)

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***Linuxové jádro a jak se v něm vyznat** (*“Jak pořádně otestovat fsck?”*)

KERN

Filip Štědronský

Co ten kernel vlastně je, čím se liší programování v kernelu od normálního kódu, jak sobě vlastní kernel postaviti a jak v něm něco opravit. Kde najít nejnovější zdrojky a kde najít pomoc, až se něco pokazí.

Sítě a bezpečnost

E-mail (*“Drahoušek zákazník.”*)

EMAIL

Kuba Maroušek, Jenda Hadrava

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, viry, phishing, BFU a kde koupit levnou viagru. Nakonec se podíváme na ne zrovna triviální grafový problém, který je v emailech skrytý.

Aplikace kryptografie * (*“6140 a184 c9a6 41f1 de99 e733 354a f451”*)

CRYPT2

Filip Štědronský, Martin Mareš

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

Filip Štědronský, Martin Mareš

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. Či 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.

Grafika a typografie

Počítačová grafika (*“Namaluj mi beránka...”*)

GFX

Martin Mareš, Jenda Hadrava, Štěpán Hojdar

Kreslení a zpracování obrazu na počítači. Co vše obnáší vykreslení obyčejné čáry, aby to bylo rychlé a pěkně vypadalo. A co teprve, když ta čára zatáčí! Vyplňování n -úhelníků a křivkou ohraničených oblastí, flood fill. Také maticové filtry pro zpracování fotek (zaostření, rozmazání), anti-aliasing a dithering. Pokud se stihne, tak navíc základy 3D vykreslování.

Barevné systémy (*“Co je na konci duhy?”*)

COLOR

Martin Mareš

O podstatě světla a barevného vidění a různých pokusech o reprezentaci barev v počítačích, fotoaparátech, televizích a podobných zařízeních. Systémy RGB, CMY(K), HSV, XYZ, Lab s jejich výhodami i neduhy. „Systém“ Pantone. Reálné kontra imaginární barvy aneb proč nejde vyfotit duha.

MetaFont, MetaPost (*“Teď ten obrázek takhle zkroutím a pak ho přeložím.”*)

MF

Jirka Setnička

Lehké nakousnutí jazyka, ve kterém můžete opravdu kreslit planimetrické obrázky, ale i třeba písma nebo piktogramy do zadání a řešení KSP. Jak vypadají CM fonty (ty, které používá \TeX) a jak se autorovi povedlo, že se z jediného „obrázku“ dá vygenerovat tlusté, tenké, rovné, skloněné, šišaté písmenko.

Typografie (*“What You See Is all What You’ve Got!?”*)

TYPO

Martin Mareš, Karry Burešová

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 T_EXnician.”*)

TEX

Jirka Setnička, Martin Mareš, Kuba Maroušek, Karry Burešová

Donald E. Knuth napsal T_EX před desítkami let proto, že mu nikdo nebyl schopen vysázet matematický text podle jeho požadavků. Od té doby se hojně používá pro sazbu nejrůznějších publikací. V této spíše praktické přednášce si ukážeme použití T_EXu od hladké sazby knihy až po zbesilosti hraničící s programováním. Pozornost věnujeme i zdrojům informací a rozdílům mezi různými dialekty T_EXu.

Teoretická informatika

Metody řešení NP-úplných problémů ** (*“I NP-úplné problémy jdou řešit.”*)

SOLVENP

Karel Tesař, Karry Burešová

NP-úplný problém je problém, pro který není znám žádný algoritmus řešící optimálně všechny instance v polynomiálním čase. To ale neznamená, že takový problém vůbec vyřešit neumíme. Podíváme se na příklady ”efektivních” exponenciálních algoritmů, parametrizace, aproximace a další pro řešení těžkých problémů.

Amortizace (*“Celek bývá daleko menší než součet částí.”*)

AMORT

Martin Mareš, (Jenda Hadrava)

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 počítadla, datové struktury založené na přebudovávání, vyhledávací stromy bez otravného vyvažování, dynamizace datových struktur, udržování historie.

Třídy složitosti *

SLOZ3

Karry Burešová, Karel Tesař

Složitost opravdu důkladně: nejrůznější třídy složitosti a vztahy mezi nimi. Vztahy mezi časem a prostorem, odstraňování nedeterminismu a Savitchova věta. Jak víme, že všechny třídy nejsou stejné: dolní odhady a věty o hierarchii. Stroje s kvantifikátory, třída PSPACE a polynomiální hierarchie. Pravděpodobnostní třídy složitosti. Orákula a neuniformní složitost.

Předpoklady: SLOZ2

Pravděpodobnostní algoritmy

PPALG

Jirka Setnička, Karel Tesař, Martin Mareš

Ukážeme si, jak v algoritmech využívat pravděpodobnost. Jak využít algoritmus, který odpovídá správně jen s pravděpodobností 0,5. Nebo jak díky náhod získat algoritmus, který je efektivní ve většině případů (Quicksort). Respektive budeme pracovat s pravděpodobnostními algoritmy typu Monte Carlo a Las Vegas a poznáme, jaký je mezi nimi rozdíl.

Vyčíslitelnost ** (*“S Halting problémem na věčné časy!”*)

VYCIS

Jenda Hadrava

Některé problémy se dají vyřešit snadno, jiné obtížněji a některé dokonce vůbec. Obecněji: Ať si vymyslíte jakýkoliv rozumný programovací jazyk, vždycky existuje problém, který se v něm nedá vyřešit. Jak se ale dokazuje, že něco nejde? Matematický pohled na výpočetní modely a univerzální stroje, rekurzivně spočetné a rekurzivní množiny a funkce. Halting problem a diagonální důkazy.

Programování s omezujícími podmínkami (*“Celé prázdniny budu plánovat a řešit sudoku.”*)

CSTR

Jirka Setnička, (Jenda Hadrava)

Trochu jiný přístup k obtížným úlohám. Některé úlohy sice vypadají, jako by se za dobu existence vesmíru nedaly vyřešit, nicméně pro rozumně velké vstupy to přesto potřebujeme. Jak backtrackovat rychleji a radostněji – backjumping, backmarking, limited discrepancy search, a jak neprobírat úplné nesmysly – hranová konzistence, konzistence po cestě, bodová konzistence.

Aplikace informatiky

Počítačová lingvistika (*“Jsou bramborové knedlíky plněné bramborami?”*)

CMPLING

Karry Burešová

Zejména motivační přednáška o počítačové lingvistice a počítačovém zpracování přirozeného jazyka. Podíváme se na vlastnosti přirozených jazyků a zaměříme se na to, jak moc komplikují jejich počítačové zpracování. Pojmenujeme odlišnosti mezi kontrolou pravopisu, automatickým překladem a konverzací s uživatelem a ukážeme si, co se zatím umí používat.

Uživatelská rozhraní (“Nákup potvrďte stiskem tlačítka s černou lebkou.”)

UX

Karry Burešová

Co mají společného automat na jízdenky, velké červené tlačítko v jaderné elektrárně a KSPÁcké webové stránky? Jsou to rozhraní, jejichž pomocí lidé něco ovládají a plní nějaké úkoly. Jak taková rozhraní navrhovat a jak je následně testovat? Proč raději nenavrhovat to, o co si lidi řeknou, a proč nevdají, že Vim by v běžném testování určitě neuspěl? A je opravdu nejdůležitější poučka „Zapomeňte na poučky“? Projdeme celý proces návrhu uživatelského rozhraní, pojmenujeme zdánlivě samozřejmé, a přece často opomíjené zásady a upozorníme na časté prohřešky.

Čárové kódy (“Jak naučit počítače číst láhve od Coly”)

BAR

Martin 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 dekodovací algoritmy plus trocha matematiky okolo zabezpečení proti chybám. Další počítačem čitelné značky: RFID, bílé křížky na asfaltu, ...

Matematické přednášky

Lineární programování *

LINPRG

Petra Pelikánová, Jenda Hadrava, Štěpán Hojdar, Martin Šerý

Jakýkoliv problém, který lze popsat soustavou lineárních nerovnic lze vyřešit v polynomiálním čase. A to metodou lineárního programování. Povíme si základy k tomu, jak lineární program vypadá, jak se vyřeší a jak různé úlohy dají popsat pomocí lineárního programu. Přednáška se může buď více zaměřit na samotnou teorii lineárního programování, a nebo na jeho aplikaci v různých algoritmických úlohách.

Optimalizace **

OPT

Petra Pelikánová

Chcete maximalizovat svůj zisk? Nebo jen potřebujete minimalizovat počet papírů na vytištění taháků daných rozměrů pro celou třídu? Zkusíme takové problémy hezky matematicky popsat, aby je za nás vyřešil počítač. Podíváme se jak (ne)mohou vypadat optimální řešení diskretních i spojitých problémů.

Předpoklady: Být kamarád s matematikou.

Logika (“Tato věta sem nepatří.”)

LOGI

Martin Mareš

Pokud budeme v životě věřit všemu, co je „přeci zřejmé“, dostaneme se brzy do potíží a v matematice to platí dvojnásob. Ale co s tím? Přírodní vědy si vymyslely verifikovatelné experimenty a matematici logiku a dokazování. Co je to výrok, co jeho důkaz a proč se axiomy nedokazují. Jenže jak si je zvolit? A jak se z toho všeho postaví celá matematika? A bude vůbec matematika někdy celá? Studená sprcha pana Gödela coby sebevražděné dovršení snahy získat dokonalý jazyk. Logika coby hra a problém líného profesora. Důkazy boží existence a neexistence.

Předpoklady: LOGI nebo její negace

Pravděpodobnost (“Většina lidí má nadprůměrný počet rukou.”)

PP

Petra Pelikánová

Toto je přednáška o základech teorie pravděpodobnosti a statistiky. Dozvíte se, co to je podmíněná pravděpodobnost, rozdělení, střední hodnota nebo rozptyl, jak se to všechno počítá a k čemu je to dobré. Součástí přednášky bude i několik zajímavých příkladů z praxe a krátký kurs přežití ve světě plném chybných statistik.

Teorie množin a matematika nekonečen * (“Kdo je nejvyšším z kardinálů?”)

TEMNO

Martin Mareš

Historie matematiky je dlážděna trampoty 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 obrnila 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 houšť kardinálů. Potenciální kontra aktuální nekonečno. Jak si pořídit přirozená čísla a jak ta reálná. Potíže s axiomem výběru.

Lineární algebra

LA

Petra Pelikánová, Jenda Hadrava

Lineární algebra původně vznikla jako elegantní prostředek k popisování geometrie lineárních útvarů (bodů, přímk, rovin, ...) v libovolněrozměrném prostoru, ale ukázalo se, že její kouzlo dosahuje daleko dál. Vektorové prostory, lineární (ne)závislost, báze, lineární zobrazení a matice, determinanty, tenzory. Konečné projektivní roviny.

Teorie (vesměs samoopravných) kódů (“f y cn rd ths, y will b gd cmprtr prgrmmr!”)

KODY

Martin Mareš

Jak komunikovat po lince, která průměrně každý k -tý bit přenesne 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 nádvkem. A jak s teorií kódů souvisí třeba čeština?

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

CPLX

Jenda Hadrava

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.

Teorie čísel a RSA * ($“2^{67} - 1 = 193\,707\,721 \cdot 761\,838\,257\,287”$)

NUT2

Martin 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.

Prvočíselné věty *

NUT3

Martin Mareš

Věty o rozložení prvočísel jsou tradičně považovány za jednu z nejmysterióznějších oblastí teorie čísel. Zde ukážeme, jak některé z nich odvodit snadným pozorováním vlastností kombinačních čísel: rozbor časové složitosti Eratosthenova síta, Bertrandův postulát („Mezi n a $2n$ je aspoň jedno prvočíslo.“), hustota prvočísel.

Fourierova transformace *

FFT

Martin Mareš, Jenda Hadrava

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 (CPLX)

Teorie nemožného * ($“Neexistence důkazu není důkazem neexistence. Dokažte.”$)

NONEX

Martin Mareš

Existenci slona v Africe snadno dokážete tím, že ho přivedete. Jak ale ukázat, že tam žádný slon není, případně že sice je, jenže ho nejde najít pomocí pravítka, kružítko a jeepu? Přímou se to dělá těžko, ale existuje spousta krásných triků, jak neřešitelnost problémů dokazovat. Nesložitelné hlavolamy, nerozvázatelné uzly, nepopsatelná čísla, neroztřetitelné úhly, nealgorithmické problémy a jiné slasti nekonstruktivní matematiky. Jak naopak ukázat, že něco existuje, aniž bychom věděli, jak to vypadá?

Úvod do klasické analýzy * ($“Dej pokoj a už mě konečně zinfinitizuj”$)

KA

Kuba Maroušek, Petra Pelikánová, Jenda Hadrava

Limita, derivace, derivace⁻¹, integrál. Z pohledu formální matematiky velmi podobné si jsou. My si ukážeme, jak jednoduše lze tyto pojmy pochopit a že vyzrát na ně není vůbec těžké.

Catalanova a Fibonacciho čísla * ($“1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, ?”$)

CAT

Martin Mareš

Kolik existuje binárních stromů? Kolika způsoby jde uzavřít 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.

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.

Lingvištika (*“Přísudek je v této větě podmět.”*)

LING

Karry Burešová, Martin 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 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.

MFF UK aneb co obnáší matfyzákem býti (*“Maminko, ptá se tatínka, kdy už budu matfyzákem?”*)

MFF

Jirka Setnička

Nezávazné povídání o Matfyzu a základním matfyzáckém folkloru. Určitě si přečteme matfyzáky sepsaný Úvod do matfyzáka a zazpíváme pár matfyzáckých písní. Zbytek už bude záležet na tom, co budete chtít slyšet.

Jak uspět v programátorských soutěžích

PSOUT

Karel Tesař

Co mám dělat, abych uspěl v soutěžích? Jak se připravovat? Na co dát důraz? Co si připravit předem a čeho si všimnout při soutěži? Jak odhadnout náročnost a které úlohy si pro řešení vybrat jako první? Je úspěch otázka talentu nebo tvrdého tréninku?

Organizování (*“Ten dělá to a ten zas tohle aneb co obnáší organizátorem býti”*)

ORG

Jirka Setnička

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.

Sbírka brouků (*“No keyboard found. Press any key to continue.”*)

BUG

Martin Mareš

Neuspořádaná přehlídka, jaké chyby kdo kdy udělal, kdy se z nich stal stal standard a jak se to komu povedlo využít. Jak některá taková chyba stála miliony dolarů, či dlouhé hodiny hledání. Očekává se přispění posluchačů.

Orientace

ORI

Martin Mareš, Jirka Setnička, Karry Burešová

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 časoměry 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.

Čaj (*“Jak vypadá odvar z nezralých pražců?”*)

TEA

Martin Mareš

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.

Email Best Practices (*“Proč si výhru nevyzvedl, když na něj blikala, svítila a pípala?”*)

EBP

Karry Burešová

Jak správně komunikovat emailem? Stejně jako papírové dopisy, i ty elektronické mají nějakou kulturu a měly by dodržovat základní štábní kulturu. Jaká pravidla dodržovat a co naopak nikdy nedělat, aby se emaily příjemci dobře četly? Jak se vyznat v stovkách emailů denně a nezbláznit se z toho? Jak psát známým, jak obchodním partnerům, a jak na konferenci? A co dělat, aby příjemce email okamžitě zahodil? Na tyto otázky si odpovíme a předvedeme i několik ukázek.

Základy první pomoci (*“Jak někomu zachránit život a jak málo k tomu stačí”*)

ZDRAV

Jirka Setnička, Karry Burešová

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.

Předmětové olympiády od A do Z

SOUT

Karry Burešová

České předmětové olympiády z pohledu soutěžícího i nezávislého pozorovatele. Jak se dostat do celostátního kola, jak (možná) dojít až do mezinárodní olympiády a která cesta vede zaručeně do pekel. Příspěvek ze strany korespondenčních seminářů, aneb zapomeňte školní znalosti, ty vám nepomůžou. Nečekejte univerzální rady, neb žádné takové neexistují, spíše vyprávění o cestě obyčejného smrtelníka olympiádním molochem.

Auto z pohledu technika (*“Co mi to vrčí pod kapotou a proč bliká ta kontrolka?”*)

CAR

Jirka Setnička

Nahlédneme do tajů starších i novějších aut. Podle zájmu se můžeme pobavit o tom, jaký je rozdíl mezi benzínovým a naftovým motorem, či proč se auta staví zrovna tak, jak se staví. Na praktické ukázké probereme (a trochu rozebereme) auto a co nejvíce si ukážeme – z pohledu běžné údržby i jednoduchých oprav. Určeno pro každého, koho čeká autoškola, nebo ho jen baví mechanika.

Technika ve sci-fi (*“Scotty, beam me up”*)

SCIFI

Jirka Setnička

Technické objevy v různých sci-fi (Star Trek/Gate/Wars i dalších) a pohled na ně z perspektivy dnešních fyzikálních znalostí. Proč se v Hollywoodských filmech ozývá ve vesmíru zvuk laserů, i když je tam vakuum, a je možné cestovat rychleji než světlo? Také možná zabrousíme do některých filozofických otázek – primární směrnice o nevměšování ve Star Treku a jiné.

Biologická evoluce (*“Ale vždyť je to jen teorie!”*)

EVOL

Filip Štědranský

Evoluční teorie je asi jedním z největších převratů v lidském náhledu na svět. Jak se na takovou věc vůbec přišlo, proč to tak dlouho trvalo a proč si myslíme, že je to pravda? Jak si to vše alespoň trochu představit? Nečekané souvislosti: proč jsou stromy vysoké, vesmír starý a atomy malé? Co je život a proč je to spíše informatický a fyzikální pojem? Jak souvisí pohlavní rozmnožování s počítačovou bezpečností, co má evoluce společného s Windows a co s halting problémem?

Předpoklady: Základní přednáška nevyžadující předchozí biologické znalosti.

Shell pro shillence (*“mkfifo p;nc -lp80<p|sed -re "s/GET /tac</s/ .*/;echo;echo& 200 OK;/q|sh|tac>p”*)

SS

Filip Štědranský

Co vše se dá napsat za pomoci unixového shellu a spřízněných utilit. Webové aplikace, šablonovací systém, síťová komunikace, OOP, ... Proč to obvykle není dobrý nápad, ale občas taky ano.

Autonomní roboti (*“Proč se točí na místě, když má jet rovně!?”*)

ROBOT

Jenda Hadrava

Jak postavit robota a jak jej naprogramovat? Povídání na pomezí virtuálního a reálného světa. Čím robot vidí okolí, jak určuje svoji pozici a jak se pohybuje? Povíme si také, kterak chytrý software může nahradit špatný hardware (a naopak).

Operační systémy (*“Mám 3GHz procesor, tak co ty Windowsy už půl hodiny dělají!?”*)

OS

Jenda Hadrava

Jak vypadá architektura dnešních operačních systémů aneb co musí programátor vědět, aby mu nepadala Wokýnka/Tučňáci. Správa procesů a vláken, plánování, synchronizace. Paměť, adresace a její přidělování. Správa souborů, filesystémy. Čemu se říká jádro a proč se spojuje s pudlem.

Jednoúčelové mikroprocesory (*“Programujeme mikrovlnku, ledničku, bagr ...”*)

MIKRO

Jenda Hadrava

Co ovládá vaše náramkové hodinky, MP3 přehrávač nebo třeba cyclocomputer. Ukážeme si, jak se programuje procesor bez OS. PIC kontra AVR. I2C, RS232, LCD displej, 7-segmentovka. Alarm a autoalarm, elektronika v autě a proč se nová auta opravují víc s notebookem než manuálně. Řízení světelné křižovatky dříve a nyní a proč není kruhový objezd lepší. Home-made řízení modelové železnice a srovnání s reálným provozem. Možná ukázky existujících zařízení.

Filosofie programovacích jazyků (*“Your programming language sucks!”*)

YPLS

Martin Mareš & Filip Štědranský

Společné filosofování nad programovacími jazyky a zejména nad tím, proč jsou všechny na draka, jen některé víc než jiné. Proč lze mít všechno na světě, jenže ne najednou. Proč je někdy lepší nechat programátora, aby si sedl na své vlastní vidle. Proč jsou některé vlastnosti jazyků na první pohled geniální, zatímco jiné (až) na ten druhý. Proč je důležitá hustota kódu a Huffmanův princip. Proč je někdy důležitější evoluce než revoluce.

Svoboda a soukromí v digitálním světě

DRIGHTS

Filip Štědranský

Kdo, proč a jak se nás o ně snaží připravit? Co o vás ví Google, Facebook, Apple či NSA a jakou nad vámi mají moc? Kdo může z vašeho telefonu vzdáleně číst a mazat data? Čím hrozí „kyberzákony“ a mezinárodní dohody? DMCA, SOPA, ACTA, TT(I)P a další. Bude Hollywood rozhodovat o tom, co (ne)smíte vidět na internetu a Microsoft o tom, jaký software (ne)smíte používat? Proč je často protizákonné upozornit uživatele na bezpečnostní chyby? A kdy váš to vše může stát život?

Lockpicking (*“Jak si odemknout, když si náhodou my (nebo soused) zapomeneme klíč :-)”*)

PICK

Jirka Setnička

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.

Létání a natáčení s dronem (*“Ne ne, to není řízená raketa, tam jen visí velký foťák.”*)

DRON

Jirka Setnička

Na co je dobré mít drona, kdy se dá reálně použít a kdy je to jen drahá hračka. Přehled o technologiích, jejich možnosti a omezení. Jak se liší armádní drony od dronů pro kameramany a nebo od dronů, které pořídíte za pár šupů v hračkářství. Také lehký výlet do legislativy – s čím smíme létat, kde smíme létat a co s tím můžeme dělat. Pokud počasí dá, tak i praktické hrátky s dronem.

Počítač bez myši (*“Může počítač trpět fobií z hlodavců?”*)

KEYB

Karry Burešová, Filip Štědronský, Martin Mareš

Ruce programátora patří na klávesnici, přesto se spousta věcí běžně dělá pomocí myši. Pojďme se podívat na programy, které s ovládáním klávesnicí počítají, a triky na ty ostatní. Spíše diskuse než přednáška, postřehy a zkušenosti všech zúčastněných jsou vítány.

Výběr vysoké školy (*“Tady prý dobře vaří!”*)

VS

Karry Burešová

Vybrat si vysokou školu může být náročný úkol. Čím se vlastně jednotlivé školy liší a čím z toho se obvykle moc nevytahují? Na co je dobré se při výběru ptát a co je spíš drobný detail? Souvisí nějak náročnost studia a vlídnost vyučujících? Povídání o rozdílech mezi školami od někoho, komu byla jedna škola málo.

Karry Burešová

Ať věřící či nevěřící, s projevy náboženství se setkáváme dnes a denně. Kde se ale taková věc vzala? Může být náboženství jen projevem nevyspělé techniky, nebo je za tím mnohem víc? Je srovnatelné pohanství dávných Slovanů a víra dnešních indiánů? A co má společného křesťanství s východními náboženstvími? Ne každý muslim je terorista, ale každý terorista je muslim. Nebo ne? A jak má člověk brát vážně křesťany, když se dohadují o blbostech? Co mají společného a čím se liší bohoslužby napříč jednotlivými monoteistickými náboženstvími i napříč jednotlivými církvemi? Pojďme si popovídat o tom i o mnohém dalším.

Vektorizace obrázků (“*Jak se rodí hroši?*”)

SVG

Petra Pelikánová

Každý hezký text se neobejde bez obrázků. Dozvíte se jak se z hlavy někoho kreativního dostane obrázek hrocha až na leták KSP. Jak a proč z náčrtku tužkou udělat vektorový obrázek a v čem se dá tvořit vektorová grafika.

Hippo::Web (“*Znovu vynalézáme kolo, ať konečně není hranaté.*”)

HIPPO

Martin Mareš

Organizátoři KSPčka rádi dělají pokusy na sobě samých, a tak při vývoji nového webu semináře pohrdli všemi vyzkoušenými webovými frameworky a napsali si potvůrku jménem Hippo. Trochu netradiční, ale v mnoha ohledech velice příjemnou. Namátkově: HTML coby datová struktura, triky s kryptografickými tokeny, kombinování programovacích jazyků a tvorba dynamických stránek, které jsou přitom skvěle cacheovatelné. Případně také nahlédnutí do dalšího zákulisí KSPčka.

Spravujeme linuxový server (“*Kolik uživatelů ti nadává, tolikrát jsi adminem.*”)

SLS

Jenda Hadrava

Povídání o tom, co se všechno může stát, když spravujete server. Od uživatele, který prozradil heslo ke svému webmailu spambotovi, přes občasné čtení logů, zálohování apod. až do situace, kdy server spadnul a musíme vyrazit na místo. Užitečné pro začínající kořeny.

Předpoklady: Virtuální svetr do virtuální serverovny.

Abecední seznam přednášek

LYK Stručný úvod do základů teorie vlkodlaků.. 1

Základní přednášky

DS2	Datové struktury pro pokročilé	1	KOMP	Od zdrojáku k programu.....	3
DS1	Datové struktury pro začátečníky	1	PARSE	Parsing čili analýza textu	2
DYNP	Dynamické programování	1	PERL	Perl	2
PHP	Dynamický web a PHP	3	HW	Principy počítačů.....	3
EVA	Evoluční algoritmy	4	THREAD	Procesy a vlákna	3
GEOM	Geometrie a počítače	2	PLX	Programování v Linuxu	3
GA	Grafy & algoritmy	1	CIS	Programování v jazyce C#.....	2
GRAFY	Grafy bez algoritmů	5	C	Programování v jazyce C	2
HASK	Haskell	3	DFS	Prohledávání do hloubky	1
REGEX	Hledání v textu	2	PYTH	Python	2
INT	Intervalové počítání	5	SHELL	Skriptování v shellu	3
ITREE	Intervalové stromy	1	SLOZ2	Složitější složitost.....	4
SQL	Jazyk SQL	3	MACHINE	Strojové učení	4
AUTO	Jazyky, gramatiky a automaty	4	NET2	Sítě II – protokoly a síťové útoky	4
KOMB	Kombinatorika	5	NET	Sítě a Internet	4
CRYPT	Kryptografie	4	TOKY	Toky v sítích	1
LOGP	Logické programování.....	3	UNIX	UNIX	3
MODEL	Modely počítačů	4	AI	Umělá inteligence.....	4
OOP	Objektově orientované programování nejen v C++	2	ZAKL	Základní algoritmy a jejich složitost	1
			NUT	Úvod do teorie čísel.....	5

Rozšiřující přednášky

AMORT	Amortizace	9	PERL6	Perl 6	6
CRYPT2	Aplikace kryptografie	8	PYTH2	Pokročilé povídání o Pythonu	7
COLOR	Barevné systémy	8	GFX	Počítačová grafika	8
CWIZ	C for wizards	6	CMPLING	Počítačová lingvistika	9
CACHE	Cache-oblivious algoritmy	8	PCRYPT	Praktická kryptografie	8
CAT	Catalanova a Fibonacciho čísla	11	PP	Pravděpodobnost	10
DS3	Datové struktury pro šílence	6	PPALG	Pravděpodobnostní algoritmy	9
DIGI	Digitální elektronika a hradla	8	CSTR	Programování s omezujícími podmínkami ..	9
EMAIL	E-mail	8	PASM	Programování v assembleru	7
FS	File systémy	7	NUT3	Prvočíselné věty	11
FFT	Fourierova transformace	11	TREES	Stromové algoritmy	6
GDB	Gdb a jiné ladicí nástroje	7	SUFF	Suffixové stromy	6
GIT	Git a jiné systémy pro správu verzí	7	KODY	Teorie (vesměs samoopravných) kódů	10
STYLE	Jak se nestat vepřem	7	TEMNO	Teorie množin a matematika nekonečen ..	10
GOLANG	Jazyk Go	7	NONEX	Teorie nemožného	11
CPLX	Komplexní a komplexnější čísla	11	NUT2	Teorie čísel a RSA	11
PRESS	Komprese dat	7	VIM	Textový editor Vim	7
LA	Lineární algebra	10	TYPO	Typografie	9
LINPRG	Lineární programování	10	SLOZ3	Třídy složitosti	9
KERN	Linuxové jádro a jak se v něm vyznat	8	UX	Uživatelská rozhraní	10
LOGI	Logika	10	VYCIS	Vyčíslitelnost	9
MAGIC	Magické algoritmy	6	TEX	TeX	9
MAKE	Make	7	KA	Úvod do klasické analýzy	11
MF	MetaFont, MetaPost	8	CPP	Černá magie v C++	6
SOLVENP	Metody řešení NP-úplných problémů	9	BAR	Čárové kódy	10
CESTY	Nejkratší a jiné cesty	6			
OPT	Optimalizace	10			

Půlnoční přednášky

CAR	Auto z pohledu technika	13	ORI	Orientace	12
ROBOT	Autonomní roboti	13	KEYB	Počítač bez myši	
EVOL	Biologická evoluce	13	SOUT	Předmětové olympiády od A do Z	13
EBP	Email Best Practices	12	RELIG	Religionistika prakticky	
YPLS	Filosofie programovacích jazyků	13	BUG	Sbírka brouků	12
FON	Fonetika	12	SS	Shell pro šílence	13
HIPPO	Hippo::Web		SLS	Spravujeme linuxový server	
PSOUT	Jak uspět v programátorských soutěžích ..	12	DRIGHTS	Svoboda a soukromí v digitálním světě ..	13
MIKRO	Jednoúčelové mikroprocesory	13	SCIFI	Technika ve sci-fi	13
LING	Lingvistika	12	SVG	Vektorizace obrázků	
PICK	Lockpicking	13	VS	Výběr vysoké školy	
DRON	Létání a natáčení s dronem		ZDRAV	Základy první pomoci	12
MFF	MFF UK aneb co obnáší matfyzákem být 12		TEA	Čaj	12
OS	Operační systémy	13			
ORG	Organizování	12			