

JARNÍ SOUSTŘEDĚNÍ KSP 2024 – 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!

Přednášky jsou členěné do třech kategorií. Na ty, o kterých si myslíme, že jsou základní a esenciální pro každého programátora, na přednášky pokročilejší, dotýkající se zajímavých oblastí informatiky, a nakonec na přednášky půlnoční, které často rozšiřují obzory až za rámec informatiky. 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 (nebo nabídku možných přednášejících, pokud je zde více jmen) a nakonec stručný obsah přednášky. Pokud u přednášky není uveden žádný přednášející, umí ji přednést většina přednášejících a jen vás nechceme unavovat přehlídkou našich jmen :-). Hvězdičky znamenají obtížnost.

Základní přednášky

Mezi těmito přednáškami jsou věci, které by měl každý začínající programátor umět. Bez pochopení většiny věcí přednášených na těchto přednáškách se budete na pokročilých přednáškách, které na ně navazují, jen obtížně chytat. Doporučujeme proto nejdříve zvládnout tyto přednášky a osvěžit si nějaký základní programovací jazyk, než se pustíte do pokročilejších věcí.

Základy programování (“*Má $x = x + 1$ řešení?*”) **ZAKL**
Úvodní trojdílná přednáška pro ty, kteří mají s programováním jen malé, nebo dokonce žádné zkušenosti. Vysvětlíme si od základů problematiku programování, jako je zápis cyklů, podmínek a funkcí, ukážeme si základní datové typy (n-tice, seznamy, slovníky), datové struktury (fronta, zásobník) a zkusíme si prakticky naprogramovat několik základních algoritmů. Vše se bude ukazovat hlavně na jazyku Python, který je jednoduchý na naučení a přesto zároveň velmi mocný. Jednotlivé přednášky se budou prolínat s přednáškami ZALG.

Základy algoritmicke, složitosti a datových struktur (“*Co by měl každý programátor znát.*”) **ZALG**
Základní vícedílný kurz algoritmů a datových struktur, který se bude prolínat se ZAKL. Jak poznat, který algoritmus je efektivnější? Přehled základních algoritmů. Co je to datová struktura a několik jejích ukázek. Vše si procvičíme na příkladech.

Grafy & algoritmy (“*Stromy, listy, lesy, pařezy, cesty, kružnice. . .*”) **GA**
Adam Jahoda, Jirka Kvapil, Kiki Prokopová, Martin Koreček, Janek Hlavatý, Honza Černý
Spousta algoritmických problémů se dá popsat pomocí teorie grafů. Ukážeme si její základy: co je to graf, jak se dá v programu reprezentovat a k čemu se dá použít. Naučíme se hledat nejkratší cestu v bludišti nebo na mapě.

Dynamické programování (“*Kampak jsem si to jenom schoval?*”) **DYNP**
Kiki Prokopová, Kristýna Petrlíková, Adam Jahoda, Honza Černý
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.

Základní programovací jazyky a techniky

Principy počítačů (“*A opravdu uvnitř počítače běhají malí trpaslíci?*”) **HW**
Jirka Kvapil, Jirka Setnička, Martin „Medvěd“ Mareš
Ukážeme si, proč programy fungují tak, jak jsme zvyklí. Co umí procesor, co dělá paměť a jak se to dá k něčemu použít. Ukážeme si nějaký program v Céčku a v Assembleru a koukneme se, kolik toho řeší Python za nás. Co dělá operační systém, jak je třeba možné, že na jednom procesoru běží najednou několik procesů. Ukážeme si, že počítače jsou překvapivě hloupá stvoření, co umí jenom základní počty, ale na programování nám to stačí.

Janek Hlavatý, Michal Kodad

Úvod do Pythonu pro ty, kteří již umí programovat v jiném jazyce. V čem se liší od ostatních jazyků a proč se v něm píše tak snadno. Proč se překládá až při spuštění, jaké výhody a jaké nevýhody to s sebou nese. Letmý úvod do balíčků aneb skoro všechno již někdo napsal za nás.

Pokročilé přednášky

Tyto přednášky by měly jednak dále rozvíjet znalosti ze základních přednášek, ale také nabízet další zajímavé programátorské techniky a technologie, které se mohou každodenně hodit.

Algoritmizace

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

ITREE

Jirka Setnička

Intervalový strom je datová struktura pracující s intervaly umožňující nám pokládat na nich dotazy třeba na největší prvek v intervalu (a to i když si dovolíme měnit hodnoty). Ukážeme si, jak tato struktura funguje, na co všechno se dá použít, jak ji aktualizovat a jak ji upravovat, aby zvládla ještě víc.

Treap

TREAP

Michal Kodad

Vyvažované stromy jsou úžasná datová struktura, ale většina normálních vyvažovaných stromů je pracné naprogramovat správně. Místo toho si řekneme, že náhoda to za nás vyváží. Sice přijdeme o pěknou worst-case složitost, ale bude se tato datová struktura lehce programovat. Navíc namísto normálních operací na vyhledávacích stromech, tak budeme moct podporovat i intervalové úpravy.

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

PARSE

Martin „Medvěd“ 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.

Hledání v textu (*“»Vyšíváme v seníku!« – kde jsem to jen viděl?”*)

TEXT

Kiki Prokopová

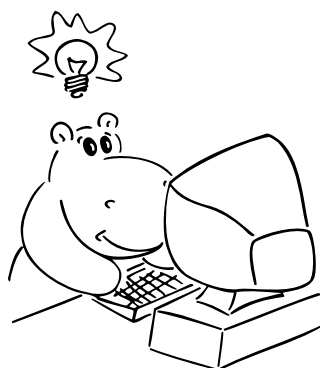
Někdy potřebujeme najít podřetězec ve velkém množství textu. Jak to udělat co nejrychleji? K tomu se nám budou hodit vyhledávací automaty, například Knuthův-Morrisův-Prattův algoritmus. Od toho dojdeme k dalším algoritmům na zpracování textu.

Datové struktury pro pokročilé * (*“Pojďme na procházku binárním lesem”*)

DS

Jirka Setnička, Martin „Medvěd“ Mareš

Přehled šikovných datových struktur, které se nevešly do ZALG. Vyhledávací stromy a různé způsoby jejich vyvažování a „ozdobení“. Hešování aneb hledáme v téměř konstantním čase. Líné datové struktury a amortizovaná složitost.



Těžké problémy *

HARD

Martin Koreček, Martin „Medvěd“ Mareš

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.

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

AMORT

Martin Koreček, Martin „Medvěd“ Mareš

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.

Další programovací jazyky a techniky

Databáze a SQL (“*SELECT something FROM knowledge LIMIT 90min*”)

DBSQL

Martin „Medvěd“ Mareš

Co je to databáze a k čemu se jí hodí a nehodí používat. Jak se s databází dohodnout jazykem SQL. Jak si pořídit tabulku, jak ji naplnit daty a jak se na ně potom ptát. Vztahy mezi tabulkami, různé druhy klíčů a operace JOIN. Pokročilejší témata: integritní omezení, indexy a transakce.

Programování v jazyce C

C

Jirka Kvapil, Jirka Setnička, Martin „Medvěd“ Mareš

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 knihovnamy jazyka C.

Procesy a vlákna * (“*Koupil jsem dalších 15 procesorů, proč je to stále stejně pomalé?*”)

THREAD

David Klement, Jirka Kvapil

Jak vypadá víceprocesorový či vícejádrový počítač a co to znamená pro programátora? Procesy, vlákna a úskalí komunikace mezi nimi aneb jak se stejným kusem paměti může pracovat více procesů. 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

Počítače, sítě, systémy

Sítě a Internet (“*Od jednoho drátu k živoucímu ekosystému*”)

NET

Jirka Setnička, Martin „Medvěd“ Mareš

Jak funguje Internet a počítačové sítě vůbec: od elektronů v drátech (fotonů v optických kabelech nebo elektromagnetických vln) přes komunikaci na jedné malé síti až ke komunikaci v celém Internetu. Vysvětlíme si rámce, pakety, MAC a IP adresy, routování v malých i ve velkých sítích. Jak to reálně funguje s IPv4 a NATem, co to jsou porty a jak se od sebe liší TCP a UDP. A na závěr radosti a strasti IPv6 (až ho konečně zavedeme).

Sítě II – protokoly (“*Jak si přečíst mailly. . . sousedovy mailly.*”)

NET2

Jirka Setnička, Martin „Medvěd“ Mareš

Volně navázání na NET aneb máme fungující síť a chceme nad ní provozovat složitější komunikaci. ICMP aneb servisní protokol Internetu, DNS a překlad doménových jmen, jednoduché textové protokoly jako je FTP, SMTP, IMAP nebo nejpoužívanější webové HTTP. U HTTP se zastavíme trochu déle – hlavičky, návratové kódy, cookie, více domén na stejné IP adrese a SSL certifikáty.

Předpoklady: Základní povědomí o počítačových sítích v rozsahu NET

Hackování webů

WEBHACK

Jirka Setnička

Praktické ukázky různých útoků na webové stránky. Od útoků na server samotný (neošetřené parametry, SQL injection, template injection) přes kradení přihlašovacích cookies pomocí XSS útoků až po přinucení udělat uživatele něco, co udělat nechce pomocí CSRF útoků. Vše si budeme prakticky ukazovat a zároveň demonstrovat, jak se proti těmto útokům bránit.

Předpoklady: Rozumět základům HTTP

Linux pro uživatele (“*Linux gives you enough rope to hang yourself.*”)

LINUX

Martin „Medvěd“ Mareš

Operační systém Linux si původně vyrobili programátoři pro sebe, a dodnes to na něm je vidět. Nenabízí tolik uživatelského pozlátka, ale dá se s ním pracovat mnohem efektivněji. Pojdme ho trochu prozkoumat. Zjistíme, že je to stavebnice složená ze spousty malých kousků, které dělají jednoduché věci a dají se kombinovat. Ovládají se pomocí příkazů, což bývá často rychlejší než klikátka. Také můžete mít svůj systém pod kontrolou a přesně vědět, co se děje uvnitř.

Linux pro správce serveru (“*Printer is on fire???*”)

LSERV

Jirka Setnička, Martin „Medvěd“ Mareš

Jak vytvořit jednoduchý Linuxový server, který poskytuje služby vaší domácnosti, nebo třeba nějaké větší síti. Co se tam hodí provozovat? Povíme o SSH, klíčích, 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 počítači.

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 periférií. Ukážeme si, jak se mikrokontroléry programují, jaké periférie 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.

Vývoj software

Systém pro správu verzí Git (*“U svatýho tučňáka, kdo sem napsal tohle? Ono to tvrdí, že JÁ?!”*)

GIT

David Klement, Jirka Kvapil, Jirka Setnička

Když se něco vyvíjí delší dobu, přijde vhod nějaký nástroj, který by uměl zjistit, kdo co přidal a proč, uměl by se vrátit k předchozí verzi nebo třeba vrátit jenom jednu změnu, co udělal kamarád před rokem. Na jeden takový, Git, se podíváme. Povíme si, jak Git ukládá změny, co jsou commity, větve, tagy a jak vypadá merge mezi větvemi. Nakonec možná předvedeme i nějaké užitečné triky: třeba hledání bugů půlením historie.

Vývoj software

DEV

Jirka Setnička, Martin „Medvěd“ Mareš

Výroba software není zdaleka jenom o programování. Pokud chceme vyvíjet větší kus softwaru ve více lidech, pojí se s tím spousta věcí – verzování, merge requesty, code review, testování a mnoho dalšího. Ukážeme si, jak může vypadat vývoj něčeho většího (ať už je to zadání KSPčka nebo třeba software pro tisíce serverů) a jaké zvyky je dobré si vypěstovat (a jaké ne). Povíme si o různých způsobech testování, o tom, jak udržet v kódu pořádek, a o dalších nástrojích, které pomáhají vyvíjet kvalitní software.

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

VIM

Martin „Medvěd“ Mareš

Odložme na chvíli své myši a pojdme 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.

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.

Praktický úvod do Jekyllu

JEKYLL

Honza Černý

Chtěl sis někdy vytvořit vlastní webovou stránku, ale ručně psát HTML a CSS tě otravuje? Zajímalo by tě, jak z Markdownu generovat pěkně vypadající, databázemi a jinou havětí nezatížené webové stránky? Ukážeme si, jak si pomocí Jekyllu vytvořit, spravovat a v neposlední řadě také publikovat webovou stránku.

Aplikace informatiky a matematiky

Počítačová lingvistika (*“Co dobrého může vzejít z propojení informatiky a lingvistiky”*)

NLP

Honza Černý

Pojďme společně nahlédnout pod pokličku počítačové lingvistiky a počítačovému zpracování přirozeného jazyka. Podíváme se na vlastnosti přirozených jazyků a povíme si, jak moc jednotlivé vlastnosti komplikují počítačové zpracování. Ukážeme si také několik praktických aplikací počítačové lingvistiky – kontrolu pravopisu, strojový překlad a vyhledávání v dokumentech podle témat atd.

L^AT_EX (*“Sázíme mrkev a odstavce”*)

LATEX

David Klement

Píšete seminárku či řešení KSP a chcete, aby výsledek vypadal k světu? Pak sáhněte po L^AT_EXu. Ukážeme si, jak málo stačí k tomu, abychom zvládli vysázat hezký dokument. Kromě toho si povíme něco málo o typografii.

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

CRYPT

Jirka Setnička, Martin „Medvěd“ Mareš

Kryptografie se zabývá šiframi, jejich konstrukcí a zejména jejich luštěním. Začneme se symetrickými a asymetrickými šiframi a jednosměrnými funkcemi. Z nich pak vybudujeme složitější kryptografické protokoly na bezpečný přenos, autentikaci a digitální podpisy. Vymyslíme dokonce, jak si hodit korunou po telefonu, a také předvedeme nerozluštitelnou šifru a dokonce to o ní dokážeme.

Unicode (*“Jaký kód má sněhulák s kudrnatými vlasy?”*)

UNI

Martin „Medvěd“ Mareš

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.

Zpracování dat (*“Bez práce nejsou koláč. . . ové grafy.”*)

DATA

Kristýna Petrlíková, 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, . . . Pojdme 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, pojdte ostatním předvést ty své.

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.

Strojové učení (*“Ať si to ten algoritmus radši vymyslí samo.”*)

ML

Michal Kodad, Honza Kotovský

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

Michal Kodad

Základy neuronových sítí: od biologického neuronu a jeho modelu – perceptronu, přes algoritmus back propagation, až po hluboké a konvoluční sítě a jejich použití na rozpoznávání obrázků.

Jak funguje ChatGPT * (*“Více dat + více výkonu = lepší systém”*)

CHATGPT

Jan Kaifer

Jak lidé přišli na ChatGPT a proč funguje. Budeme si povídat hlavně o zajímavostech a nebudeme si kazit den matematikou a jinými technikaliami. Zmíníme, co je to strojové učení a jak fungují neuronové sítě. A co za triky museli lidé vymyslet, než jsme dokázali natrénovat modely, které pohánějí ChatGPT. Přednáška nevyžaduje předchozí znalosti.

Herní algoritmy (*“Když nemáte na to, abyste vyhráli šachový turnaj. . . ”*)

AIGAME

Jirka Setnička, 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 si tedy, jak hru zanalyzovat.

SAT a jeho aplikace *

APPSAT

Kristýna Petrlíková, Michal Kodad

Představíme si problém splnitelnosti výrokových formulí, tzv. SAT. Podíváme se, jak lze různé těžké inforatické problémy formulovat v řeči SATu. Vybrané problémy mohou být například z oblastí barvení grafů, plánování, různých logických hádanek či verifikace kódu. Nebudou chybět praktické ukázky v Pythonu.

Programování s omezujícími podmínkami (*“Neumím řešit sudoku, ale znám pravidla!”*)

CSP

Michal Kodad

Často se dostaneme do situace, kdy řešíme NP-úplný problém (sudoku, barvení mapy n barvami) a musíme napsat spoustu stejného (boilerplate) kódu, který vůbec nezávisí na daném problému. Zde přichází na pomoc programování s omezujícími podmínkami (CSP). Pomůže nám popsat podstatu problému a přitom se nezajímat o to, jak se daný problém bude řešit, ani zbytečnými implementačními detaily. Povíme se o základních technikách, které používají CSP řešiče, a popíšeme nějaký problém v řeči CSP.

Přírodou inspirované algoritmy (*“Kolik mravenců potřebujeme, aby spočítali NP těžký problém?”*)

NATURALG

Honza Černý

Příroda je nádherná a celá tisíciletí se jí inspirujeme. Kolik už inspirovala spisovatelů, básníků a malířů. Nyní jsou na řadě programátoři. Když si nebudeme vědět rady s nějakým těžkým problémem, zkusíme nenápadně opisovat od přírody.

Matematické přednášky

Grafy bez algoritmů

GRAFY

Ondra Chwiedziuk, Martin „Medvěd“ Mareš

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.

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

KOMB

Ondra Chwiedziuk, Martin „Medvěd“ Mareš

Pokusíme se vybudovat kombinatoriku intuitivně. Co nejvíce se vyhneme počítání se vzorci, vystačíme si s elegantními úvahami. Kromě základních technik si ukážeme, jak nám mohou pomoci rekurence a jak se úlohy dají převádět mezi sebou. Procvičíme na spoustě příkladů.

Matematická analýza * (*“Matfyzák se snaží integrovat do společnosti.”*)

MA

David Klement

V rychlosti si zavedeme derivace a integrály a následně si ukážeme, k čemu všemu se hodí. Hledání minima funkce, kleslení hladkých křivek, aproximace. Výpočet délky spirály nebo objemu roztodivných těles. Využití ve fyzice: pohyb s odporem vzduchu, zamrzání ledu na rybníce a mnoho dalšího.

Průlet lineární algebrou

LIN1

Adam Jahoda, David Klement, Janek Hlavatý, Honza Černý, Ondra Chwiedziuk

Podíváme se, čím se to ta lineární algebra vlastně zabývá. Řekneme si, co jsou matice, jak se s nimi počítá a k čemu jsou dobré. Seznámíme se s pojmy jako těleso, vektor a vektorový prostor, představíme si jejich zajímavé vlastnosti a uvedeme je do různých souvislostí.

Logické programování (*“Mohu být svým vlastním dědečkem?”*)

LOGP

Kristýna Petrlíková, Jirka Kvapil, Honza Černý

Což takhle projednou neříkat počítači, jak má věci počítat, ale jenom mu zadat podmínky, které má výsledek splňovat? Neprocedurální programování vychází přesně z této myšlenky. Podíváme se na programovací jazyk Prolog, který vychází z formální logiky. Zjistíme, které problémy se v něm neobyčejně zjednoduší a které naopak programování promění v noční můru. Pokud jsi milovník rekurze, budeš u této přednášky nejspíš skoro celou dobu spokojeně vrnět.

Funkcionální programování (*“Když vám život dá funktory, udělejte si monádu”*)

FUNP

Jirka Kvapil

Ve funkcionální programování popisujeme výpočet programu pouhými funkcemi a jejich vyhodnocováním. Zmíníme historii od které přejdeme k rozumným jazykům, konkrétně k Lispu. Zjistíme proč proměnné nelze upravovat a proč to nejsou proměnné v pravém slova smyslu. Vysvětlíme co jsou to funkce vyšších řádů a jak nám šetří spoustu práce. Dále ukážeme další velice užitečné nápady jako makra nebo pattern matching. A pokud zbyde čas vše zakončíme několika funktory ze kterých uděláme monády.

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

OOP

???

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 čisté virtuální (abstraktní) metody. Jak se liší OOP ve statických (C++, C#, Java) a dynamických (Python) jazycích. Jak programovat objektově i bez podpory jazyka, třeba v Cěčku.

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

Jazyky, gramatiky a automaty

AUTO

Kristýna Petrlíková, Adam Jahoda, Jirka Kvapil, Martin „Medvěd“ Mareš

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?

Teorie informace * (*“Proč zazipovaný zip není menší než původní?”*)

TEOINF

Ondra Chwiedziuk

Zadefinujeme si, co je to informace, jak jí můžeme měřit a jaké to má důsledky. Z počátku se prokoušeme pravděpodobností, řekneme si, co je to entropie a jak souvisí s bitem informace. Následně si povíme, co je to kód a jak nejlépe můžeme zakódovat informaci. Vysvětlíme si Huffmannovo kódování a, pokud zbude čas, i aritmetický kód a Ziv-Lempelův rekurenční kód.

Úvod do teorie čísel a RSA (*“Pošli mi zprávu, tady máš klíč.”*)

RSA

Ondra Chwiedziuk, Martin „Medvěd“ Mareš

Řekneme si, čím se zabývá teorie čísel a vysvětlíme si její neznámější aplikaci: šifru RSA. Vybudujeme teorii kolem dělitelnosti, řekneme si základní větu aritmetiky, vysvětlíme si Eukleidův algoritmus, Bézoutovu rovnost, Eulerovu funkci a důležité vlastnosti prvočísel. Když budeme znát všechny tyto pojmy, můžeme se pustit do vysvětlení, proč vlastně RSA funguje. Na závěr si řekneme slabiny RSA a proč už se dneska přestává používat.

Ostatní přednášky

WTF is HFT (*“Kolik bude dnes stát Bitcoin?”*)

TRADE

Jan Kaifer

Úvod do obchodování na burze. Jak funguje burza a podle čeho se určuje, za kolik si můžete koupit jedno euro nebo akcii Googlu? Od koho to vlastně kupujete? A jaké problémy řeší firmy, od kterých si to kupujete?

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

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. Jak se jazyky vyvíjejí a jak se navzájem ovlivňují. Kde jsme přišli k pravidlům a jaký je jejich smysl. Existují synonyma? Proč je jazyk nejednoznačný a proč je to dobře. Jak se na jazyk dívá matematik a jak se na matematiku dívají lingvisté. Jak vzniklo písmo? A jak otazník? Jak zapsat zachrochtání a jak třeba mlasknutí &c.

GPS (*“Čekám na signál. . .”*)

GPS

David Klement

GPS je vstutku magická technologie, a ačkoli se to nezdá, využívá jedny z nejdůležitějších výsledků moderní vědy. Vysvětlíme si, na jakém principu GPS funguje, a proč by to jinak nešlo.

Teorie množin (*“Jablka a hrušky se dají nejen sčítat, ale třeba i násobit.”*)

TEMNO

Martin „Medvěd“ Mareš

Základoškolský přístup „množina je kupříkladu miska jablíček“ nabízí spoustu otázek: Když jablíčka přesuneme do sáčku, bude to stále tatáž množina? A co když kousek jablíčka ukousneme? V rámci této přednášky se pokusíme o vybudování teorie množin od základů (rozuměj axiomů) a to v duchu Zermelo-Fraenkelovském. Pak uvidíme, jak na teorii množin vystavět zbytek matematiky.

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

TYPO

???

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.

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

TEA

Martin „Medvěd“ 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.

První pomoc a zdravotní péče pro začátečníky (*“Je to jednodušší než to vypadá”*)

ZDRV

Adam Jahoda, Kiki Prokopová, Jirka Setnička, Janek Hlavatý

Pobavíme se o základech první pomoci. Jak se postarat o člověka v bezvědomí? Jak stav člověka určit. A k čemu má smysl volat záchranku? Ukážeme si, jak málo stačí k tomu někomu zachránit život. Pokud bude zájem, tak si můžeme i něco vyzkoušet prakticky.

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.

Bezpečnost umělé inteligence (*“Jak zabránit počítačům, aby k nám poslali z budoucnosti nahého Arnolda S.?”*)

AISAFETY

Honza Černý

Umělá inteligence se k nám blíží každým dnem. S ní se ale blíží i spousta reálných rizik. Podívejme se společně na reálná nebezpečí, která s umělou inteligencí přicházejí a jestli jim dokážeme zabránit.

Geometrie algoritmicky

ALGEO

Ondra Chwiedziuk

Taky tě nikdy nebavily geometrické úlohy ve škole a na Matematické Olympiádě? Nebaví tě hledat těžiště, používat větu o obvodovém a středovém úhlu, provádět kruhovou inverzi? Pokud jsi odpověděl alespoň na jednu z těchto otázek ano, pak je tato přednáška přímo pro tebe. Existuje totiž algoritmus, který dokáže všechny tyto úlohy vyřešit. Začneme klasickou analytickou geometrií, načež si řekneme, jak se dá efektivně počítat s polynomy, aby nám daly námi žádaný výsledek.

Proč jít na stáž a jak se tam dostat? (*“Pohovory jsou rozbité, tak toho pojďme využít.”*)

STAZ

Jan Kaifer

Stáže jsou skvělé, získáte nové kamarády a vaši kariéru to možná pomůže více než vysoká škola, a přesto nejsou v Čechách tak oblíbené. Problém ale často spočívá v tom, že se na stáž není jednoduché dostat. Probereme, jak fungují stáže a pohovory v prestižních firmách jako Google, Meta nebo Jane Street. Ukážeme si, podle čeho se u pohovorů hodnotí, na co se zaměřit a jak jednoduše získat body navíc.

Abecední seznam přednášek

LYK Stručný úvod do základů teorie vlkodlaků.. 1

Základní přednášky

DYNP	Dynamické programování..... 1	ZALG	Základy algoritmizace, složitosti a datových struktur..... 1
GA	Grafy & algoritmy..... 1	ZAKL	Základy programování..... 1
HW	Principy počítačů..... 1		
PYTH	Python..... 2		

Pokročilé přednášky

AMORT	Amortizace..... 3	THREAD	Procesy a vlákna..... 3
DBSQL	Databáze a SQL..... 3	CSP	Programování s omezujícími podmínkami.. 5
DS	Datové struktury pro pokročilé..... 2	C	Programování v jazyce C..... 3
FUNP	Funkcionální programování..... 6	LIN1	Průlet lineární algebrou..... 6
GRAFY	Grafy bez algoritmů..... 5	NATURALG	Přírodou inspirované algoritmy..... 5
WEBHACK	Hackování webů..... 3	APPSAT	SAT a jeho aplikace..... 5
AIGAME	Herní algoritmy..... 5	ML	Strojové učení..... 5
TEXT	Hledání v textu..... 2	GIT	Systém pro správu verzí Git..... 4
ITREE	Intervalové stromy..... 2	NET2	Sítě II – protokoly..... 3
CHATGPT	Jak funguje ChatGPT..... 5	NET	Sítě a Internet..... 3
AUTO	Jazyky, gramatiky a automaty..... 6	TEOINF	Teorie informace..... 6
KOMB	Kombinatorika..... 5	VIM	Textový editor Vim..... 4
CRYPT	Kryptografie..... 4	TREAP	Treap..... 2
LSERV	Linux pro správce serveru..... 3	HARD	Těžké problémy..... 2
LINUX	Linux pro uživatele..... 3	AI	Umělá inteligence..... 5
LOGP	Logické programování..... 6	UNI	Unicode..... 4
MA	Matematická analýza..... 6	VSCODE	Visual Studio Code..... 4
MCU	Mikrokontroléry..... 4	DEV	Vývoj software..... 4
NEURO	Neuronové sítě..... 5	TRADE	WTF is HFT..... 6
OOP	Objektově orientované programování..... 6	DATA	Zpracování dat..... 5
PARSE	Parsing čili analýza textu..... 2	LATEX	L ^A T _E X..... 4
NLP	Počítačová lingvistika..... 4	RSA	Úvod do teorie čísel a RSA..... 6
JEKYLL	Praktický úvod do Jekyllu..... 4		

Půlnoční přednášky

AISAFETY	Bezpečnost umělé inteligence..... 7	STAZ	Proč jít na stáž a jak se tam dostat?..... 7
GPS	GPS..... 7	ZDRV	První pomoc a zdravotěda pro začátečníky.. 7
ALGEO	Geometrie algoritmicky..... 7	TEMNO	Teorie množin..... 7
LING	Lingvištika..... 7	TYPO	Typografie..... 7
PICK	Lockpicking..... 7	TEA	Čaj..... 7