

JARNÍ SOUSTŘEDĚNÍ KSP 2025 – 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, dšs 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. Všechny přednášky ze základní kategorie na soustředění budou, proto nemusíš rozdělit všechny body v této kategorii. Hlasováním nám dej vědět, jak moc už látku z dané přednášky umíš: 0 bodů = všemu rozumím a přednášku nepotřebuji, 20 bodů = vše je pro mě nové.

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 algoritmizace, složitosti a datových struktur (*“Co by měl každý programátor znát.”*)

ZALG

Honza Kaifer

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

Honza Kotovský, Jirka Kvapil, Kačka Vomelová, Vojta Káně

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.

Základní programovací jazyky a techniky

Principy počítačů (*“A opravdu uvnitř počítače běhají malí trpaslíci?”*)

HW

Jirka Setnička, Martin „Medvěd“ Mareš, Jirka Kvapil, Jirka Kalvoda

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.

Kristýna Petrlíková, Jirka Kalvoda

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

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

DS

Jirka Setnička, Martin „Medvěd“ Mareš, Jirka Kalvoda

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.

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

GEOM

Jirka Setnička, Jirka Kalvoda

Jak pomocí počítače vyřešit věci, které na papíře řešíme metodou „kouknu a vidím“? Jak najít konvexní obal množiny bodů, jak zjistit, jestli je bod uvnitř mnohoúhelníku, jak spočítat obsah nějakého (ne)konvexního obrazce a další. Na spoustu věcí se nám bude hodit algoritmus zametání roviny, nejprve přímkou a později možná i něčím složitějším (třeba až budeme počítat Voroného diagramy).

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

ITREE

Jirka Setnička, Jirka Kalvoda

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.

Evoluční algoritmy * (*“Trocha mutace nikdy nikoho nezabila, ne?”*)

EVO

Honza Kotovský

Lidstvo má dlouhou historii kradení nápadů od přírody. Tudíž není žádným překvapením, že jakmile přišel Darwin s teorií evoluce, informatici už v koutě kuli pikle, jak můžou tuto krásnou teorii zneužít. Tak přišli na svět evoluční algoritmy, způsob jak nechat sílu přírody pracovat ve váš prospěch. Brzo se ukázalo, že je to až překvapivě mocná metoda, která dokáže najít velmi solidní aproximaci i pro velmi komplexní problémy.

Neuronové sítě * (*“Jak si postavit vlastní neuronovou síť a naučit jí číst”*)

NN

Honza Kotovský, Vladimír Sklenář

Umělá inteligence je dneska zastřešující termín pro až děsivé množství věcí. Někde v kořenech tohoto spletitého lesa jsou neuronové sítě, na kterých stojí mocné stromy velkých modelů jako třeba všeznáma Chat-GPT. Naneštěstí většina AI tutoriálů vám dá všemocné nástroje, aniž by vám ukázaly co se v nich skutečně děje. Pojďme tedy společně postavit neuronku jen za pomoci trocha násobení a sčítání! A možná ji naučíme i něco dělat a ne jen smutně existovat. . .

Hledání v textu * (*“Jak najít jehlu v kupce sena a přitom se nebodnout?”*)

TEXT

Vladimír Sklenář, Jirka Kalvoda, Vojta Káně

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.

Parsování textu (*“ $1+2*4 = 12$ ”*)

PARSE

Standa Lukeš

Často potřebujeme načíst nějaký složitý textový vstup: matematický výraz, zdroják, konfigurační soubor, . . . Ukážeme si, jak takové textové formáty analyzovat (neboli parsovat), aniž bychom v nich zabloudili: jak si napsat rekurzivní parser od základu i jak použít automatictější nástroje založené na regulárních výrazech a gramatikách.

Těžké problémy *

HARD

Martin „Medvěd“ Mareš, Jirka Kalvoda

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

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

AMORT

Martin „Medvěd“ Mareš, Jirka Kalvoda

Spousta algoritmů je mnohem rychlejší, než jak na první pohled vypadají. Šikovný způsob, jak takové chování zkoumat, je amortizovaná časová složitost. Předvedeme několik trochu překvapivých příkladů amortizace: dvojková a jiná počítadla, datové struktury založené na přebudovávání, vyhledávací stromy bez otravného vyvažování, dynamizace datových struktur, udržování historie.

Jedním z nejmocnějších algoritmických nástrojů jsou rekurzivní funkce – tedy takové, které volají samy sebe. Ukážeme, k čemu může být rekurze užitečná a jak se v ní neztratit. A kdy je naopak lepší se rekursi vyhýbat.

Další programovací jazyky a techniky

Programování v jazyce C

C*Jirka Setnička, Jirka Kvapil, Kačka Vomelová*

Jazyk C je tu s námi již více než 50 let, ale stále patří k velmi používaným jazykům, které se hodí umět. C najde využití od low-level programování přes zpracování videa až po psaní linuxového kernelu, zejména pokud má být něco rychlé. Představíme si datové typy a běžné programové konstrukce, vysvětlíme si základy práce s ukazateli a také se seznámíme se standardními knihovnami jazyka C.

Pokročilý Python (*“I can’t comprehend this comprehension.”*)

PYTH2*Jirka Kalvoda*

Povídání o tom, co nám Python nabízí, aby se v něm hezky programovalo, které zajde o trošku dále než základní kurz, určené zejména těm, co už v pythonu programovali před soustředěním. Jak psát *pythonicky* aneb `enumerate`, `zip`, `comprehensions` či generátory. Jak se neztratit v balíčcích a jak si vytvořit vlastní. Co za pasti skýtá Python a jak přesně fungují pythonní třídy. Jak se dívat na pythonní objekty z pythonu a případně jak za běhu programu modifikovat asi tak cokoliv a proč to možná nechceme dělat.

Předpoklady: Základy Pythonu

Programovací jazyk Julia (*“Python je pomalý a v Céčku grafy kreslit nechcete”*)

JULIA*Standa Lukeš*

Julia je relativně mladý dynamický skriptovací jazyk zaměřený na zpracování dat, strojové učení a matematiku. I když má trochu jinou syntaxi než Python nebo Rko, je jimi inspirovaný; a dokonce můžete přímo volat Pythoní a Rkové knihovny. Na rozdíl od Pythonu je ale silně typový, což umožňuje mít efektivní JIT (kompilaci do strojový kód). Jinak má, co byste od moderního jazyka čekali: funkce, `lambdy`, vlastní operátory, objekty, ... a přičetný správce knihoven.

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

FUNP*Jirka Kvapil, Jirka Kalvoda, Standa Lukeš*

Ve funkcionální programování popisujeme výpočet programu pouhými funkcemi a jejich vyhodnocováním. Zmíníme historii od které přejdeme k rozumným jazykům, konkrétně k Haskellu. 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`.

Pracujeme a programujeme v shellu (*“„MakáSH!“ zavolal přísným hlasem programátor. A on makal. . .”*)

SHELL*Vojta Káně*

Ovládání počítače pomocí textových příkazů je v některých kruzích stále populární pro svou efektivitu a snadnou automatizaci. Podívejme se na něj trochu z blízka počínaje klasickým POSIX shellem a BASHem. Pak se vydejme alternativními směry – Nu, Xonsh. A bylo-li libo nějaké objekty? Máme tu PowerShell.

Webové stránky

WEB*Honza Kaifer*

Naučíme se základy tvorby webových stránek pomocí HTML a CSS. Ukážeme si, jak strukturovat obsah stránky, jak jej stylovat a jak přidat základní interaktivitu pomocí JavaScriptu. Během přednášky si každý vytvoří svou vlastní webovou stránku, kterou si bude moci dále rozvíjet. Dozvíte se také, jak stránky publikovat na internet, aby je mohli vidět i ostatní.

Webové doplňky

EXT*Honza Kaifer*

Naučíme se, jak vytvořit jednoduchý doplněk (rozšíření) pro webový prohlížeč. Ukážeme si, jak doplňky fungují, jak mohou interagovat s webovými stránkami a co všechno s nimi lze dělat. Během přednášky si vytvoříme vlastní doplněk, který nám usnadní práci s webem nebo přidá nějakou zábavnou funkcionalitu do prohlížeče. Vyzkoušíme si, jak doplněk nainstalovat a jak ho sdílet s ostatními.

Asynchronní programování (*“Kdo čeká, ten se dočká. A tím spíš, pokud čeká víckrát naráz.”*)

ASYNC*Jirka Kalvoda, Vojta Káně, Standa Lukeš*

Jistě jste se setkali s programy, co například čekali na vstup od uživatele. Co když ale chceme čekat na více věcí současně? Například krom odpovědi od uživatele může dorazit něco po síti, z jiného programu nebo ze souboru. Na této přednášce si ukážeme různé způsoby, jak tyto problémy řešit. Od aktivní smyčky přes low-lever rozhraní linuxího kernelu – rodiny `poll`, přes programování za pomoci `callbacků` po asynchronní programování v Pythonu.

Počítače, sítě, systémy

Základy sítí (*“Jak si postavit Internet?”*)

NET

Jirka Setnička, Martin „Medvěd“ Mareš, Jirka Kvapil, Jirka Kalvoda, Vojta Káně

Jak fungují počítačové sítě: od elektronů v drátech (nebo fotonů v optických kabelech) 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 jak postupně zavádíme IPv6.

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

NET2

Jirka Setnička, Martin „Medvěd“ Mareš, Jirka Kalvoda

Volné 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, Standa Lukeš

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

SERVER

Jirka Setnička, Martin „Medvěd“ Mareš, Jirka Kalvoda, Vojta Káně

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 periferie typicky obsahují a jak je používat ke komunikaci s okolním světem. Něco si vyzkoušíme i prakticky na STM32.

Předpoklady: Hodí se základní znalost jazyka C.

Relační databáze (*“Šéf se ve firmě mění obrok, programátoři každých deset let. Data zůstávají. . .”*)

RDBMS

Vojta Káně, Standa Lukeš

Napsat od základy třebas jen rezervační formulář pro knihovnu není vůbec jednoduché. Jak zařídit, že se data neztratí po nečekaném výpadku napájení? Jak nejlépe předejít vytvoření rezervace pro dva čtenáře naráz, kteří si zažádali ve stejnou chvíli? A jak později připojit mobilní aplikaci? Už desítky let je častou odpovědí na otázky výše relační databáze, která převezme péči o naše data a my už je jen servírujeme uživatelům. Přístup je to velmi oblíbený a nějakou databází dnes nabízí prakticky každý webhosting. Podívejme se, s čím vším nám tedy databáze pomohou, jak se v nich data strukturují do tabulek a propojují klíči a jak je naopak posléze můžeme dotazovat jazykem SQL.

Vývoj software

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

GIT

Jirka Setnička, Jirka Kvapil, Jirka Kalvoda

Když se něco vyvíjí delší dobu a/nebo v týmu, přijde vhod nějaký nástroj, který umí ukládat různé verze zdrojového kódu, zjistit kdo co přidal a proč, k nějaké verzi se vrátit nebo třeba vrátit jenom jednu změnu, kterou udělal kamarád před rokem. Git je dnes nejpoužívanější takový nástroj. Povíme si, jak ukládá změny, co jsou commity, větve, tagy a jak vypadá merge mezi větvemi. Nakonec zkusíme předvést i nějaké užitečné triky: třeba hledání bugů půlením historie.

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

VIM

Martin „Medvěd“ Mareš, Jirka Kalvoda

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 prokatě efektivní. Základní příkazy, práce s regulárními výrazy, makra, kouzla. Vimovité ovládání jiných programů, třeba webového prohlížeče.

Aplikace informatiky a matematiky

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

CRYPT

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, Standa Lukeš

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

Vladimír Sklenář, Standa Lukeš

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 to prakticky v Pythonu na reálných datech, které si umíte veřejně stáhnout.

Předpoklady: PYTH

Buněčné automaty a Game of Life (*“Čtverečkový svět, co není Minecraft”*)

LIFE

Martin „Medvěd“ Mareš

Game of Life je dvojrozměrný svět, ve kterém se buňky vyvíjí podle průzračně jednoduchých pravidel. Už desítky let v tomto světě objevujeme další a další zajímavé jevy. Tak do něj také nahlédneme, prozkoumáme souvislosti s evoluční biologii i s algoritmy. Též uvidíme, jak Život zapadá do obecnějšího světa buněčných automatů.

Chipy a identifikátory (*“Aneb jak platit mámy kreditkou aniž by o tom věděla”*)

CHIP

Honza Kotovský

Bezkontaktní čipy dneska zastupují nemalé množství plastů, které každodenně používáme. Identifikace na školních obědech, lítačka na MHDčko, věrnostní karty megařetězců, bezkontaktní platby v obchodech i jako náhražka klíčů do našich domovů. Jak ale funguje tato technologie a jaké jsou její limity?

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

BAR

Martin „Medvěd“ Mareš

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

Komprese dat

ZIP

Standa Lukeš

Přehled základních kompresních algoritmů: triviální algoritmy (RLE), statistické metody (Huffmanovo a aritmetické kódování), slovníková komprese (LZ77, LZ78, LZW), Burrowsova-Wheelerova transformace (BZIP). Různé kódování dat, aby standardní kompresní algoritmy byly šťastné (prediktory, delty, bit spit).

Lineární programování prakticky

LINPR

Ondra Chwiedziuk

V této přednášce si ukážeme, jak optimalizovat počet upečených rohlíků a buchet, aby pekárna měla co největší profit. Naučíme se reálný problém linearizovat a následně vyřešit pomocí solveru v Pythonu. Také si řekneme, jaké problémy můžeme tímto způsobem vyřešit.

Matematické přednášky

Lineární algebra (nejen) pro začátečníky

LIN1

Ondra Chwiedziuk

Těleso, vektor, matice, Gaussova eliminace, vektorový prostor, lineární zobrazení, charakteristický polynom, vlastní číslo, diagonalizace. Všechny tyto pojmy si osvětlíme na úvodní přednášce z lineární algebry. Ukážeme si, co tyto pojmy znamenají, jak s nimi pracovat a nakonec, jak je aplikovat.

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

LOGP

Kristýna Petrlíková, Jirka Kvapil

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.

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

KOMB

Martin „Medvěd“ Mareš, Jirka Kalvoda

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.

Martin „Medvěd“ Mareš

Řekneme si, čím se zabývá teorie čísel a vysvětlíme si její nejznámější 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.

Aplikace matematické analýzy

MAPL

Vojta Lančarič

Nejprve si povíme o základních pojmech matematické analýzy, jako je limita, derivace funkce a integrál, abychom se poté mohli podívat na jejich využití. Povíme si o Taylorově polynomu, který se hodí k aproximaci funkce, nebo o tom, jak hledat minimum a maximum funkce. Pokud zbyde čas, podíváme se i na některá využití ve fyzice.

Grafy bez algoritmů

GRAFY

Ondra Chwiedziuk, Jirka Kalvoda

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.

Ostatní přednášky**Manimování** (“Jak si udělat animaci fraktálu?”)

MANIM

Vladimír Sklenář

Chtěli by jste si udělat vlastní animaci průchodu vašeho BFS-ka? Nebo si napsat kód na simulaci fraktálů? Ukážeme si základní animační techniky v manimu uvidíte jak generujeme KSP-reels pro Instagram. Naučíme se vektorovat obrázky a přidáme pár typů pro animátory.

Předpoklady: PYTH

Programátoři versus realita (“Jennifer Null ze Scunthorpe by mohla vyprávět . . .”)

FALSE

Martin „Medvěd“ Mareš

Sbírka představ o světě, kterým většina programátorů věří, ale vlastně nejsou pravdivé. Jsou všechny dny stejně dlouhé? A všechny minuty? Kolik znaků má jeden znak? Čím jsou oddělené comma-separated values? Kde ve Zlíně je tř. Tomáše Bati č.p. 202? Kolikrát že? Má každý občan EU příjmení? A jméno? A právě jedno rodné číslo? Kolik zavináčů může obsahovat e-mailová adresa? A pokud zbyde čas vše zakončíme několika funktory ze kterých uděláme monády.

Půlnoční přednášky

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

Lingvistika (“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.

Quo vadis, češtino? (“Muž žena skopce dokopce si píská.”)

CZLANG

Martin „Medvěd“ Mareš

O češtině časů minulých, současných i (potenciálně) budoucích. Kde se čeština vzala a jak souvisí s ostatními evropskými jazyky. Co jsme během staletí získali, o co jsme přišli a o co nejspíš brzy přijdeme. A je to škoda? Vykrádáme cizí jazyky a ony zase nás. Kolik je češtin a která je ta správná? Proč už Jan Hus spílal Pražanům? Pojďme navrhnout pravopisnou reformu!

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

TEMNO

Martin „Medvěd“ Mareš, Jirka Kalvoda

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.

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

TEA

Martin „Medvěd“ Mareš, Jirka Kalvoda

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.

Rubikova kostka bez algoritmů

RUBIK

Ondra Chwiedziuk

Už jste někdy zkoušeli složit Rubikovu kostku? A nešlo vám to? Našli jste si návod na internetu, ale byla to jen náhodná směť písmenek? Tak si ukážeme, jak ji složit bez toho. A také si ukážeme, jak Rubikova kostka souvisí s algebrou.

Jak Vojta cestoval a proč byste měli taky (*“Pomoci se na cestách dostane každému, kdo se chová slušně.”*) INTERRAIL Vojta Káně

Na dnešní napjatý svět není lepšího léku, než se po něm vypravit podívat a zjistit, že tráva je všude stejně zelená, koláče různé leč vždycky dobré a lidé milí a ochotní. Já si tuto medicínu dávkuji od počátku střední školy a na přednášce vám ji předepíšu též. Nenavštívil jsem žádné romanticky divoké kraje, neukážu, jak zdolat extrémní přírodu. Na ověřených příkladech ale představím, jak se dá za pár korun a za pár chvil procestovat Evropa.

Rozebrat, vyčistit a vdechnout život (*“Smetiště či poklad – dva pohledy na tutéž hromadu harampádí”*) FIXME Vojta Káně

Cena elektroniky je stále menší, jenže to i její životnost. O biologické rozložitelnosti ani nemůže být řeč a tak se skládky plní a naše peněženky vyprazdňují. Přitom k notebooku, který s člověkem procestoval celý svět a zažil s ním příjemné i náročné chvíle, si jeden snadno vybuduje citový vztah. A vdechnout mu pak znovu život, to je pocit, jako bych sám byl Stvořitel. Mohu pár takových úspěšných příběhů přednést a dát pár obecných dobrých rad, ale hlavně přivezu náradí a trochu know-how, takže pokud máte doma hardware v terminálním stádiu, přivezte ho a zkusíme mu dát ještě šanci.

Evoluce bez algoritmů

EVOB

Standa Lukeš

O tom jak funguje evoluce IRL - jakým způsobem život nevznikl "náhodou". A jak nám znalost těchto principů umožňuje dělat "reverzní inženýrství" živých věcí.

Abecední seznam přednášek

LYK	Stručný úvod do základů teorie vlnění... 1
------------	--

Základní přednášky

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

Pokročilé přednášky

AMORT	Amortizace..... 2	MANIM	Manimování..... 6
MAPL	Aplikace matematické analýzy..... 6	MCU	Mikrokontroléry..... 4
ASYNC	Asynchronní programování..... 3	NN	Neuronové sítě..... 2
LIFE	Buněčné automaty a Game of Life..... 5	PARSE	Parsování textu..... 2
BAR	Čárové kódy..... 5	PYTH2	Pokročilý Python..... 3
DS	Datové struktury pro pokročilé..... 2	SHELL	Pracujeme a programujeme v shellu..... 3
EVO	Evoluční algoritmy..... 2	FALSE	Programátoři versus realita..... 6
FUNP	Funkcionální programování..... 3	JULIA	Programovací jazyk Julia..... 3
GEOM	Geometrie a počítače..... 2	C	Programování v jazyce C..... 3
GRAFY	Grafy bez algoritmů..... 6	REC	Rekurze..... 3
WEBHACK	Hackování webů..... 4	RDBMS	Relační databáze..... 4
TEXT	Hledání v textu..... 2	NET2	Sítě II – protokoly..... 4
CHIP	Chipy a identifikátory..... 5	GIT	Systém pro správu verzí Git..... 4
ITREE	Intervalové stromy..... 2	VIM	Textový editor Vim..... 4
KOMB	Kombinatorika..... 5	HARD	Těžké problémy..... 2
ZIP	Komprese dat..... 5	UNI	Unicode..... 5
CRYPT	Kryptografie..... 4	RSA	Úvod do teorie čísel a RSA..... 6
LIN1	Lineární algebra (nejen) pro začátečníky... 5	EXT	Webové doplňky..... 3
LINPR	Lineární programování prakticky..... 5	WEB	Webové stránky..... 3
SERVER	Linux pro správce serveru..... 4	NET	Základy sítí..... 4
LOGP	Logické programování..... 5	DATA	Zpracování dat..... 5

Půlnoční přednášky

TEA	Čaj..... 6	CZLANG	Quo vadis, češtino?..... 6
EVOB	Evoluce bez algoritmů..... 7	FIXME	Rozebrat, vyčistit a vdechnout život..... 7
INTERRAIL	Jak Vojta cestoval a proč byste měli taky.. 7	RUBIK	Rubikova kostka bez algoritmů..... 7
LING	Lingvistika..... 6	TEMNO	Teorie množin..... 6