

JARNÍ SOUSTŘEDĚNÍ KSP 2022 – 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 daemologie 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 algoritmizace, 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 ukávek. Vše si procvičíme na příkladech.

Grafy & algoritmy (“*Stromy, listy, lesy, pařezy, cesty, kružnice. . .*”) **GA**

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, pochopíme základní princip, jak funguje navigace podle GPS a mnohé další. Vše si procvičíme na konkrétních příkladech.

Dynamické programování (“*Kampak jsem si to jenom schoval?*”) **DYNP**

Ondra Sladký, Jirka Kalvoda, Vojta Káně, Vašek Šraier, Martin Koreček

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 Setnička, Vojta Káně, Vašek Šraier

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 Asembleru 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.

Python **PYTH**

Jirka Setnička, Vojta Káně, Honza Černý, Vašek Šraier

Jak programovat v Pythonu a jak v něm „nepsat Cčko“. 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.

Vojta Káně, Honza Černý, Vašek Šraier

Slycháte v programátorských kruzích o objektech, třídách, dědičnosti a metodách, ale nemáte pořádně jasno v tom, co to přesně znamená? Představíme si objektově orientované programování (OOP) – techniku, která často pomáhá psát větší programy tak, aby se o nich snáze přemýšlelo. Ukážeme si základní principy a příklady použití. Čím se liší OOP v různých programovacích jazycích? Kdy je lepší se objektům vyhnout?

Předpoklady: Základy programování

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

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***Jirka Setnička, Jirka Kalvoda, Eliška Vítková, Vojta Káně, Honza Černý, Vašek Šraier*

Často se stane, že máme velký objem textu (knihu, všechny dokumenty na našem disku, celý internet), ve kterém opakovaně hledáme krátké řetězce (slova či sousloví). Pak se vyplatí předpočítat si pro text pomocnou datovou strukturu, která nám hledání urychlí. V našem případě touto strukturou bude sufixové pole, které nám umožní vyhledávat v čase logaritmickém k délce prohledávaného textu. Ukážeme si i další vyhledávací algoritmy, které se vyplatí třeba pokud naopak hledáme hodně dlouhá „slova“. To se hodí na trochu překvapivých místech – například se pomocí hledání v textu dají řešit některé úlohy o stromech.

Intervalové stromy * (*“Já bych ty intervaly nejradši. . . dal do stromu!”*)**ITREE***Ondra Sladký, Jirka Kalvoda, Honza Černý, Martin Koreček*

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.

Datové struktury pro pokročilé * (*“Pojďme na procházku binárním lesem”*)**DS***Kuba Pelc, Honza Černý, Jirka Setnička, Jirka Kalvoda, Vašek Šraier, Vojta Káně, Martin Koreček*

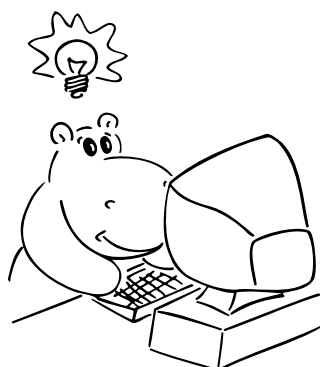
Přehled šikovných datových struktur, které se nevešly do ZALG. Vyhledávací stromy a různé způsoby jejich vyvažování a „ozdobení“. Hešování aneb hledáme v téměř konstantním čase. Líné datové struktury a amortizovaná složitost.

Geometrie a počítače (*“Nerušte mé kruhy! (ani jiné kvadriky)”*)**GEOM***Jirka Setnička, Jirka Kalvoda, Martin „Medvěd“ Mareš, Martin Koreček*

Základní algoritmy pro řešení geometrických úloh – zametání roviny, 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.

Toky v sítích * (*“Když je v grafu povodeň, těsní?”*)**TOKY***Jirka Setnička, Jirka Kalvoda, Eliška Vítková, Martin „Medvěd“ Mareš*

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.



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 vypořádávat v praxi.

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

AMORT

Kuba Pelc, Jirka Kalvoda, Martin „Medvěd“ Mareš, Vašek Šraier, Martin Koreček, Ondra Sladký

Spousta algoritmů je mnohem rychlejší, než jak na první pohled vypadají. Šikovní 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

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

THREAD

Jirka Setnička, Vašek Šraier

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

Programování webových aplikací (*“Vlastní blog či wiki za 20 minut?”*)

WEBAPP

Vašek Šraier

Pojďme si vytvořit vlastní webové stránky poháněné Pythonem. Začneme s ničím a skončíme s funkčními webovými stránkami. Ukážeme si vše potřebné – domény, server, HTML a CSS, nějaký JavaScript. U ničeho nebudeme zabíhat do hloubky (na to nemáme čas), ale pokusíme se stihnout vše potřebné, abyste mohli pokračovat v případě zájmu sami. Přednáška je polopraktická a vše bude možné si vyzkoušet. Nebudeme se ale moc zdržovat tím, že to někomu nefunguje. To až po přednášce.

Předpoklady: Python, hrubá představa o HTML a troška gitu také neškodí

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

SQL

Vojta Káně, Martin „Medvěd“ Mareš

Představíme si SQL, jazyk databází. Ukážeme si základní příkazy i práci o kus složitější. Jak databázi dát data a jak se na ně potom ptát. K čemu se hodí složený dotaz a klíčové slovo JOIN. Jak si data skupinkovat pomocí GROUP a kdy radši použít PARTITION. Dnes už SQL databáze typicky umí i mnoho rozšíření, tak si třeba ukážeme jak do ní naházet JSON dokumenty a použít ji jako dokumentovou databázi.

Programování v jazyce C

C

Jirka Setnička, Vojta Káně, Martin „Medvěd“ Mareš, Vašek Šraier

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.

Jazyk Go

GOLANG

Jirka Setnička, Vojta Káně

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 vracující libovolný počet hodnot nebo třeba vestavěná podpora Unicode a vestavěná hashovací tabulka. Také se silně dbá na coding-style pro zajištění snadné čitelnosti programů.

Iterátory (*“Nekonečný cyklus přes prvočísla...”*)

ITER

Jirka Sejkora

Iterátory jsou hojně využívanou konstrukcí, která se pod různými názvy a podobami nachází ve většině moderních programovacích jazyků. Ukážeme si jak vyrobit typické funkce, co s nimi pracují a přiblížíme se tak funkcionálnímu programování. Taky se podíváme na jejich podobu v různých jazycích, ukážeme si lambda funkce, a na závěr si možná vyřešíme úlohu z KSP-Z na jeden řádek kódu.

Počítače, sítě, systémy

Webové stránky

WWW

Vojta Káně

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

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

NET

Jirka Setnička, Vojta Káně, Martin „Medvěd“ Mareš, Vašek Šraier

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 a síťové útoky (“*Jak si přečíst maily. . . sousedovy maily.*”)

NET2

Jirka Setnička, Vojta Káně, Martin „Medvěd“ Mareš, Vašek Šraier

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, kterého se zastavíme trochu déle – hlavičky, návratové kódy, cookie, více domén na stejné IP adrese. A pokud zbude čas, využijeme zranitelnosti některých protokolů a provedeme síťový útok.

Předpoklady: Základní povědomí o počítačových sítích v rozsahu NET

Od zdrojáku k programu (“*Před spuštěním program přeložte. Stačí třikrát podělně?*”)

KOMP

Martin „Medvěd“ Mareš, Vašek Šraier

Mezi programem ve vašem oblíbeném přičetném programovací jazyce, který jste právě dopsali, a tranzistory uvnitř vašeho procesoru leží obrovské území obývané překladači, JITy, knihovničky, operačními systémy, loadery a jinými bájnými bytostmi. Pojďme zjistit, co jsou zač a co všechno s programem provádějí. Co udělá kompilátor za nás a co musíme naopak udělat my za něj.

Operační paměť * (“*Tahle příhrádka říká, kde je příhrádka, která říká, kde hledat byte 0xdeadbeef*”)

MEM

Vojta Káně, Vašek Šraier

Spustit i triviální program dá spoustu práce. A co když chce používat knihovny? A dynamicky alokovat paměť? A co když jich je víc? A ne úplně důvěryhodné? Všechny tyto problémy se lidstvo pokusilo vyřešit mnoha hardwarově podporovanými triky s pamětí. V této hře se hraje o nanosekundy, takže vše musí zůstat co nejjednodušší. Tato přednáška se nesnaží představit nový koncept, na který byste jinde nenarazili, ale uceleně shrnout téma tradičně rozdrobené přes mnoho přednášek na příbuzná témata. Společně vystopujeme cestu mallocu od zavolání této knihovní funkce až ke konkrétním buňkám paměti.

UNIX (“*UNIX gives you enough rope to hang yourself.*”)

UNIX

Jirka Setnička, Vojta Káně, Honza Černý, Vašek Šraier

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 je ovládání prostřednictvím textových příkazů často efektivnější než klikátka? Jaké to je mít svůj systém pod kontrolou a „vidět mu pod ruce“? V čem spočívá moc textových souborů?

Linuxový server (“*Chci provozovat vlastní linuxový server, ale nevím jak*”)

ADMIN

Jirka Setnička, Vojta Káně, Vašek Šraier, Martin „Medvěd“ Mareš

Na co se mi hodí vlastní server a jak ho provozovat? Domácí server nebo něco sedícího v cloudu? A když už ho mám, tak jak ho zabezpečit a jaké věci se mi na něm hodí provozovat? Budeme si povídat o SSHčce, klíčích, šifrování, systemd, Apache a Nginxu, mailech, DNS, Let’s Encrypt, zálohování a všem dalším, co nás bude zajímat. Ideálně prakticky na nějaké virtuálce. Pokud bude zájem, můžeme zabrousit i do provozování serverů a služeb ve větším aneb Kubernetes a další hračky.

Předpoklady: Základní znalost Linuxu.

Docker (“*Stavíme služby jako kostičky*”)

DOCKER

Jirka Setnička, Vojta Káně

Docker umožňuje snadno vytvořit kontejnery – třeba nějaké aplikace a všech jejích závislostí – a tyto kontejnery jednoduše distribuovat a spouštět. Docker se hodí ke snadné instalaci věcí s mnoha závislostmi nebo naopak stejné aplikace na mnoho serverů. Ukážeme si, jak si Dockerový kontejner vyrobit, jak z něj dostat porty a do něj naopak nějaké úložiště a kam kontejner umístit. Na závěr si ukážeme, jak Docker provozovat ve větších clusterech (Kubernetes), aneb když chceme distribuovat zátěž na mnoho strojů a mít redundanci.

Vývoj software

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

GIT

Jirka Setnička, Jirka Kalvoda, Vojta Káně, Martin „Medvěd“ Mareš, Vašek Šraier, Jirka Sejkora

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 púlením historie.

Testování a kvalita softwaru

QA

Vašek Šraier

Výroba software není zdaleka jenom o programování. Ukážeme si, jak psát kód tak, aby nejenom fungoval, ale aby vyzařoval krásu a pokoj všem programátorům dobré vůle. Povíme si o různých způsobech testování, o tom, jak udržet v kódu pořádek, a dalších nástrojích, které pomáhají vyvíjet kvalitní software.

Předpoklady: Znat aspoň jeden přičetný programovací jazyk

Aplikace informatiky a matematiky

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

AIGAME

Jirka Setnička

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.

Vykreslování ve hrách a OpenGL (*“Namaluj mi stádo beránek. . .”*)

GL

Kuba Pelc

Spíše popularizačně-naučný pohled do vnitřností vykreslování herních engineů, shrnutí konceptů v grafických API jako je OpenGL a jak se kombinují do jednoho celku, který dělá něco zajímavého. Jak se to dělá, aby světla svítila, věci se leskly, aby v lese byly temné stíny a proč je to všechno naprosto špatně. Co všechno umí grafické karty a jak se toho využívá při optimalizaci her. A kdo ví, možná nějaký ten engine i za živa rozpitváme. . .

Shadery (*“Malujeme kódem.”*)

SHADER

Kuba Pelc

Programování nemusí být jen o terminálech a webech. Ukážeme si, jak i z několika málo řádků kódu může vzniknout něco hezkého a barevného. Shadery jsou krátké programky, které každému pixelu obrázku přiřadí nějakou barvu. Podíváme se s jejich pomocí do hlubin fraktálů a z trošky náhody uděláme mraky nebo vodní hladinu. Přednáška bude spíše praktický workshop.

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

TEX

Jirka Setnička, Jirka Kalvoda, Martin „Medvěd“ Mareš

Donald E. Knuth napsal TEX 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 TEXu od hladké sazby knihy až po zbesilosti hraničící s programováním. Pozornost věnujeme i zdrojům informací a rozdílům mezi různými dialekty TEXu.

Kryptografie (*“Gbgg 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.

Komprese dat (*“Jnm idln kpln j nstlčtln.”*)

ZIP

Jirka Setnička, Martin „Medvěd“ 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).

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

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.

Matematické přednášky

Grafy bez algoritmů

GRAFY

Kuba Pelc, Ondra Sladký, Jirka Kalvoda, Eliška Vítková, Vojta Káně, Vašek Šraier

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

Martin „Medvěd“ Mareš

Středoškolská kombinatorika bývá otravnou „zoologií“ plnou pojmů a vzorečků. My si místo toho ukážeme přístup k počítání založený na elegantních úvahách. Základní triky s faktoriály a kombinačními čísly, sčítání konečných a občas i nekonečných řad a rekurentní rovnice. Také zajdeme na návštěvu k Dlouhému a Širokému a za poněkud zmatenou šatnářkou.

Průlet lineární algebrou

LIN1

Kuba Pelc, Martin „Medvěd“ Mareš

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.

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

NONEX

Martin „Medvěd“ 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? Přímo 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 teorie čísel

NUT

Martin „Medvěd“ Mareš

Co a k čemu je teorie čísel. Počítání modulo n , Euklidův algoritmus na nalezení největšího společného dělitele 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.

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.

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

ZDRAV

Jirka Setnička, Vašek Šraier

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.

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

PICK

Jirka Setnička, Honza Černý

Jak fungují dnešní zámky, co jsou to stavítka a jak vlastně fungují klíče. A jak se pomocí jednoduchých nástrojů dají využít výrobní nedokonalosti zámků k jejich odemčení. Použití planžet, napínáků, praktické ukázky odemykání, nastínění technik bumpingu a dalších postupů, jak se dostat přes zamčené dveře.

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.

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

LING

Martin „Medvěd“ Mareš

Převážně nevážené 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.

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

TYPO

Martin „Medvěd“ Mareš

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.

Orientace

ORI

Martin „Medvěd“ Mareš

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

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.

Abecední seznam přednášek

LYK Stručný úvod do základů teorie vlnodlaků.. 1

Základní přednášky

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

Pokročilé přednášky

AMORT	Amortizace..... 3	THREAD	Procesy a vlákna..... 3
AISAFETY	Bezpečnost umělé inteligence..... 5	C	Programování v jazyce C..... 3
BAR	Čárové kódy..... 5	WEBAPP	Programování webových aplikací..... 3
DS	Datové struktury pro pokročilé..... 2	LIN1	Průlet lineární algebrou..... 6
DOCKER	Docker..... 4	SHADER	Shadery..... 5
GEOM	Geometrie a počítače..... 2	NET	Sítě a Internet..... 4
GRAFY	Grafy bez algoritmů..... 5	NET2	Sítě II – protokoly a síťové útoky..... 4
AIGAME	Herní algoritmy..... 5	SQL	SQL databáze..... 3
TEXT	Hledání v textu..... 2	GIT	Systém pro správu verzí Git..... 4
ITREE	Intervalové stromy..... 2	NONEX	Teorie nemožného..... 6
ITER	Iterátory..... 3	QA	Testování a kvalita softwaru..... 4
GOLANG	Jazyk Go..... 3	TEX	T _E X..... 5
KOMB	Kombinatorika..... 5	HARD	Těžké problémy..... 3
ZIP	Komprese dat..... 5	TOKY	Toky v sítích..... 2
CRYPT	Kryptografie..... 5	UNIX	UNIX..... 4
ADMIN	Linuxový server..... 4	NUT	Úvod do teorie čísel..... 6
KOMP	Od zdrojáku k programu..... 4	GL	Vykreslování ve hrách a OpenGL..... 5
MEM	Operační paměť..... 4	WWW	Webové stránky..... 3
PARSE	Parsing čili analýza textu..... 2		

Půlnoční přednášky

TEA	Čaj..... 7	TEMNO	Teorie množin..... 6
LING	Lingvistika..... 6	TYPO	Typografie..... 6
PICK	Lockpicking..... 6	ZDRAV	Základy první pomoci..... 6
ORI	Orientace..... 6		