

JARNÍ SOUSTŘEDĚNÍ KSP 2011 – Seznam Přednášek

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 a nakonec stručný obsah přednášky.

Povinně dobrovolné přednášky

Základy programování (“*Má $x = x + 1$ řešení?*”) **ZAKL**

Lukáš Lánský, David Marek

Úvodní několikadílná přednáška pro ty, kteří mají s programováním malé nebo dokonce žádné zkušenosti. Od základů si vysvětlíme problematiku programování a představíme programovací jazyk Python, který je jednoduchý na naučení, ale přitom není jen dětskou hračkou a je používán ve spoustě důležitých projektů. Ukážeme si základní datové typy (n-tice, seznamy, slovníky) a datové struktury (fronta, zásobník) a základní algoritmy a programátorské techniky, bez kterých se žádný programátor neobejde.

Základní algoritmy **ZALG**

Základní výbavou každého informatika jsou různé standardní algoritmy, zde si ukážeme ty nejdůležitější z nich: Třídící algoritmy včetně vnějšího třídění. Vyhledávání v polích, hledání mediánu, resp. k -tého největšího prvku v lineárním čase. Vyhodnocování výrazů, předzpracování vstupních dat.

Grafy & algoritmy I (“*Pojďme si hrát s obrázky*”) **GA1**

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. Topologické třídění grafů a kreslení grafů jedním tahem.

Datové struktury pro začátečníky **DS1**

Jak si ukládat data natolik šikovně, abychom je nejen neztratili, ale také našli dříve, než si pro nás přijde Smrť. Klasické struktury jako pole, seznamy, vyhledávací stromy (vyvážené, AVL, a - b , splay), haldy (binární a obecně regulární) a v neposlední řadě hashování.

Algoritmy a jejich složitost (“*Čím menší je časová složitost algoritmu, tím větší je složitost kódu.*”) **SLOZ**

Problém, algoritmus a program. Časová a paměťová složitost problémů i algoritmů. Složitost rekurzivních algoritmů, složitost v průměrném případě.

Dokazování **DOKAZ**

Karel Král

Ověřit správnost programu můžeme buď spuštěním algoritmu na opravdu velkém množství různorodých vstupů, nebo rozumnou úvahou, která mimo vší pochybnost existenci vstupu, na který by program odpověděl špatně, vyloučí.

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

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.

Jak vypadá řešení (“*Jen jeden bod, když jsem napsal 18 stránek?*”) **SOL**

Jak má správně vypadat řešení KSP? Na co si dát pozor, co je úplně špatně a za co organizátoři strhávají body a sobě vlasy. Přednáška, která by mohla pomoci i mnohým déle aktivním řešitelům.

Úvod do programátorských soutěží (“*Za pár let chci třítmat v rukou pohár!*”) **SOUT**

Karel Tesař, Martin Böhm

Tipy a triky, jak uspět v programátorských soutěžích a olympiádě. Čeho si všimnout při vymýšlení algoritmů a na co si dávat pozor při samotné implementaci. Jakých soutěží se na střední škole můžete účastnit, kde se dají získat zkušenosti a kde se naopak dají vyhrát velké ceny.

Algoritmizace

Složitější složitost * **SLOZ2**

Martin Böhm

Trochu hlouběji o složitosti: amortizovaná časová složitost, dolní odhady, nedeterministické výpočty a třída NP, NP-úplné problémy a příklady redukci.

Předpoklady: SLOZ

Datové struktury pro pokročilé *

DS2

Martin Böhm

Důmyslnější datové struktury: trie, splay stromy, BB- α stromy; geometrické struktury pro lokalizaci bodů v rovině; binomiální a Fibonacciho haldy, leftist haldy a 2-3 haldy. Též několik přátelských randomizovaných datových struktur: skip listy a treapy.

Grafy & algoritmy II

GA2

Martin Böhm, Lukáš Lánský

Pokročilejší grafové algoritmy: union-find problem, hledání minimální kostry, párování v grafech, testování vícenásobné souvislosti a silné souvislosti.

Intervalové stromy * (*“Intervaly, intervaly, intervaly a jak na ně.”*)

ITREE

Karel Tesař

Probereme vše, co by měl správný programátor vědět. Co jsou a k čemu slouží intervalové stromy. Jaké se pomocí nich dají řešit úlohy. Probereme množství stromů v 1D, 2D a více-D, statické i dynamické. Ukážeme implementaci podle Fenwicka, binárně indexované stromy i rekurzivní konstrukci. Jak se dají elegantně napsat za 5 minut. Jako bonus hromada příkladů.

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

TOKY

Martin Böhm, Lukáš Lánský

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)

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

REGEX

Lukáš Lánský

Vyhledávání čehokoliv ve velkém množství textu: Prostá vylepšení hledání hrubou silou: Karp-Rabin, Boyer-Moore. A algoritmy chytřejší: Morris-Pratt, Knuth-Morris-Pratt, Aho-McCorasicková. Konečné automaty prakticky, regulární a „regulární“ výrazy.

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

GEOM

Martin Mareš

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.

Umělá inteligence ve hrách (*“Když nemáte na to, abyste vyhráli šachový turnaj . . .”*)

AIGAME

Pavel Veselý, David Marek

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.

Programovací jazyky a techniky**Programování v jazyce C**

C

Pavel Veselý, Martin Böhm

Datové typy jazyka C, programové konstrukce, základy práce s ukazateli. Seznámení se standardními knihovny jazyka C.

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

CIS

Pavel Veselý, Lukáš Lánský

C# je moderní objektově orientovaný jazyk, který za deset let svého bouřlivého vývoje dostal do vínku některé funkcionální rysy. Mimo popisu základních konstrukcí si projdeme zajímavé části dotnetí Base Class Library.

Úvod do knihovny STL (*“Jak napsat binární vyhledávací strom za 2 minuty?”*)

STL

Karel Tesař, Pavol Rohár

STL je sada knihoven pro C++, která je velmi užitečná a hojně používaná při programátorských soutěžích. V STL je implementována řada základních algoritmů a datových struktur jako je například třídění, binární hledání, operace pro haldu, nafukovací pole, set, který lze používat stejně jako binární strom, atd.

Python (*“import antigravity”*)

PYTH

David Marek

Základy programování v Pythonu, syntaxe, datové typy, funkce, třídy, moduly. Kolik existuje verzí Pythonu, jaké novinky na nás čekají v Pythonu 3. Výhody interaktivního interpretu.

Psaní webů v Pythonu (*“Po této přednášce nebudete chtít PHP ani vidět”*)

WEBPY

David Marek

Python je jazyk, který si získal srdce mnoha vývojářů webů. Díky frameworkům jako jsou Django, Web2py nebo Pylons už není psaní stránek stejně úmorné jako s PHP. Bude vybrán jeden framework a na něm předvedeno jak lehce se dají psát stránky. Výhody MVC přístupu, šablon, ORM knihoven. Pro ty, co nemají Python hosting, si ukážeme Google App Engine.

Předpoklady: PYTH

Základy webu

WWW

Tomáš Maleček, Lukáš Lánský

Úvod do servírování informací skrze World Wide Web. Co znamená `http://`, jaký je rozdíl mezi HTML a XHTML a na co se hodí znát kaskádové styly.

Jemnosti webu

WWW2

Lukáš Lánský

Web je velké téma a točí se kolem něj spousta pojmů v různém stádiu vyprázdněnosti. Syndikační formáty jako Atom a iCal beze sporu existují a fungují, do jaké míry to ale můžeme prohlásit o entitách typu SEO? Jsou mikroformáty minulost, přítomnost, nebo budoucnost? Spoustu času můžeme věnovat rozprávění o nových rysech HTML5.

PHP

PHP

David Marek, Tomáš Maleček

Úvod do problematiky vytváření dynamicky generovaných webových stránek. Syntaxe jazyka PHP, netypovost, (ne)pole. Jak psát stránky, za které se nemusím stydět, objektové programování, použití MVC. Nejpoužívanější frameworky. Zamícháno s XHTML, JavaScriptem, CSS a možná i (My)SQL.

Předpoklady: WWW

Jazyk XML a související technologie (“`<xml style="vesele"/>`”)

XML

Tomáš Maleček, Lukáš Lánský

Povíme si, co je to jazyk XML, jak vznikl a k čemu je dobrý. Zároveň probereme nástroje pro práci s XML dokumenty, jejich načítání (parsování), generování, validaci atd. Z pokročilejších technik pak ukážeme vyhledávání v XML pomocí jazyka XPath a transformace XML dokumentů pomocí XSLT. Zbyde-li čas ukážeme si i několik konkrétních aplikací (XHTML, WML, SVG, ODF ...).

Prolog * (“*Detektivem za 90 minut.*”)

LOGP

Jitka Novotná

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 * (“`fac n = product [1..n]`”)

HASK

Lukáš Lánský, Martin Böhm

Funkce v obvyklých (imperativních) programovacích jazycích nejsou funkce v matematickém slova smyslu: mají „vedlejší efekty“, projevují se na běhu programu i jinak, než svou návratovou hodnotou. Ve funkcionálních jazycích, mezi něž Haskell patří, mají takové chování zakázáno. K čemu to?

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

PERL

Martin Mareš

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 složité 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. Malé ochutnání Perlu6, jazyka (snad už nepříliš vzdálené) budoucnosti.

Programování v UNIXu (“*Segmentation fault. Core dumped.*”)

PUNIX

Martin Mareš

Jak se programuje v operačním systému, který si programátoři navrhli pro sebe? Vyměníme klikací vývojová prostředí za stavebnici se spoustou kostiček. Textový editor Vim, překladač GCC a otrokář zvaný make. Jak zjistit, co program doopravdy dělá, i když běží hodinu – core dumpy, GDB, valgrind, oprofile. Jak programovat ve více lidech a nezbláznit se z toho – diff, patch, Git.

Matematické přednášky

Diskrétní matematika (“*O Dlouhém, Širokém a šatnářce*”)

DM1

Pavel Veselý, Karel Tesar

Úvodní minikurs diskrétní matematiky (to je opak matematiky spojité, čili mimo jiné kombinatorika). Seznámení s relacemi a jejich vlastnostmi. Dozvíte se také něco o uspořádaných, nezávislých a jiných množinách. S pomocí kombinatoriky možná vyřešíme problém zmatené šatnářky. Hallova věta nám pomůže určit, jestli má cenu snít o perfektním párování.

Lineární algebra (“*Vektorový prostor je místo, kde žijí vektory.*”)

LA1

Martin Böhm

Lineární algebra vznikla jako formalizace geometrie a tuto souvislost si ukážeme. Popíšeme vektorové prostory, které se skládají z vektorů. Jaké operace s nimi umíme provádět a co všechno musí splňovat? Kdy jsou vektory závislé a kdy nezávislé? Co je to lineární kombinace, obal a generátor? Co je to dimenze vektorového prostoru a jaké má souvislosti s předchozími pojmy?

Kryptologie (*“Gbgg arav zbp gnwan mcenin.”*)

CRYPT

Martin Mareš, Karel Tesař

Kryptologie 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 jak si hodit korunou po telefonu. Předvedeme nerozluštitelnou šifru a dokonce to o ní i dokážeme.

Logika (*“Tato věta sem nepatří.”*)

LOGI

Lukáš Lánský

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 sebevražedné dovršení snahy získat dokonalý jazyk. Logika coby hra a problém líného profesora. Důkazy boží existence a neexistence.

Předpoklady: LOGI

Teorie kombinatorických her (*“Život je jen hra... Jakou má vyhrávající strategii?”*)

GAME

Pavel Veselý, Lukáš Lánský

Rozličné kombinatorické hry se zápalkami, kamínky, barvičkami či grafy a jejich abstrakce. U některých si ukážeme výherní či obranné strategie, u některých dokážeme, že sice příslušná strategie existuje, ale nikdo ji ve skutečnosti nezná. Teorie pana Conwaye: jak pozicím přiřazovat čísla, popř. jiné symboly. Piškvorky a jejich zobecnění na hypergrafy.

Teorie grafů

GRAF

Lukáš Lánský, Martin Böhm

Grafy, které vás na střední škole neučili! Žádné osy a funkce, jsou to města a silnice. Jednoduché, představitelné objekty. Často je vidáme v algoritmech, ale i z čistě matematického úhlu pohledu jsou zajímavé. Ukážeme si nečekaná použití grafů v geometrii, teorii čísel i jinde. Na grafech si také představíme dualitu – vlastnost, kterou mají snad všechny matematické pojmy společnou. Vše jednoduše a srozumitelně i pro ty, kdo o grafech slyší prvně.

Rovinné grafy

ROG

Pavel Veselý, Lukáš Lánský

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

Fraktály

FRAC

Lukáš Lánský

Nekonečně složité obrázky, které jdou ale často matematicky popsat velmi jednoduše. Necháme počítač nakreslit fraktální kapradí, vločku a mořského koníka. Na programu jsou také fraktály v komplexní rovině a v Pascalově trojúhelníku. Ukážeme si, jak vlastně fraktály vznikají a naprogramujeme si jednoduché kreslírko Mandelbrotovy množiny.

Hardware, operační systém a ostatní

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

UNIX

Pavol Rohár

Kamarád u černobílého textového okna září blahem. Chcete poznat, proč? Jak UNIX vznikl, k čemu je dobrý a k čemu třeba není. UNIXová filosofie. Kouzlo scriptů. Kouzlo speciálních souborů. Kouzlo propojování programů. Kouzlo nechtěného. UNIX byl napsán v C a C vzniklo pod UNIXem.

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

HW

Martin Mareš, Tomáš Maleček

Historie a vývoj počítačových technologií a současných architektur. Co je to procesor, jak se programuje (instrukční sady) a jak se chová. Přehled tříd procesorů (skalární a vektorové, RISC). Jak procesor komunikuje s okolím (čipová sada). Nezbytné periferie: vnitřní a vnější paměti, vstupní a výstupní zařízení.

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

NET

Lukáš Lánský

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. Protokoly rodiny TCP/IP, síťové topologie (a proč Internet vlastně nemá žádnou), internetworking. Pár taktů hudby budoucnosti: IPv6, multicasting, přenos v reálném čase atd.

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

GFX

David Marek

Kreslení a zpracování obrazu na počítači. Souřadnice (rovinné, prostorové i barevné) a jejich transformace. Základní grafická primitiva: body, úsečky, kružnice, elipsy, Bézierovy křivky a jejich rasterizace. Vyplňování n -úhelníků a křivkou ohraničených oblastí, flood fill. Pár triků navíc: maticové filtry, anti-aliasing a dithering. Grafické formáty a komprese obrázků. Základy trojrozměrného promítání a vykreslování scény.

Martin 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. Probereme základy TEXu, jeho využití při psaní řešení KSP, ale třeba také povídek, ročenky, slajdů pro prezentaci, zpěvníku nebo rovnou not.