

Cvičení Programování I

Cvičící: **Pavel Surynek, KTIML,**
pavel.surynek@seznam.cz
Semestr: **Zima 2005/2006**
Kroužek: **Matematika/59**
Rozvrh: **Pátek 10:40-12:10 (učebna K2)**

Stručné poznámky ke cvičení ze 2.12.2005

1. Organizační záležitosti. Všichni cvičící sepsali témata k zápočtovým programům. Zde je sbírka všech možných témat.

Surynek

U každého tématu je vyznačena obtížnost číslem od 1 do 5, přičemž 1 znamená nejlehčí a 5 nejtěžší. Radu témat lze dělat libovolně lehce i libovolně těžce (címž mám na mysli rozpetí obtížnosti zhruba 3 až 5).

Hry - piskvorky, karty, sudoku, dama, šachy (u her pro dva hráče lze navrhnout inteligentního protihráče - piskvorky, dama; u her pro jednoho hráče lze udelat třeba napovědu dalšího tahu - sudoku atd.) [3-5]. Procházení bludisté [3-5], Rubikova kostka [3-5]. Inteligentní kalendáře [1-3], textové editory [3-5], formátování textu [2-4], prohlížeče souborů se zalamováním řádku [2-4]. Simulator procesoru [3-5], interpret jednodušeho jazyka [3-5], kalkulačka vyřazova [3-5], kalkulačka na velká čísla [3-5], kreslička na funkce (lze prostorově i plošně) [3-5]. Grafika - raytracing kulíček [3-5], vektorové kresličko [3-5], bitmapové kresličko (trochu technické) [2-5], zobrazování 3D objektu [3-5]. Simulace - gravitace [3-5], aerodynamika [4-5], galaktické srážky [5], korist/predator [3-5]. Neuronové síťe - učící algoritmy, rozpoznávání písmen [3-5]. Šifrování - desifrování textu zasíťovaného pomocí neznámé šifry [2-5]. Fraktály - kreslení fraktálu [1-3]. Hadání dalšího členu posloupnosti [1-5]. Kulečník [3-5] (autor nápadu je tuším pan Čekal).

Valla (pozn. kráceno)

Variace na buněčné automaty, prohlížeč textových (či volitelně i jiných) souborů, textová adventura, textový dungeon, generator hmatu na kytaru, univerzální kalendář, nejruznější backtrackové hlavolamy [pozn. třeba lišák], grafický viewer, sazba matematického vzorečku, polyrootsolver.

Kryl

Knihovna práce s dlouhými čísly, Gaussova eliminace, Logik - MasterMind, jednoduchý grafový algoritmus, Sudoku, rozsypaná křížovka, řešení dvoutažky, vyhodnocování sportovních soutěží (losování švýcarského systému v šachách, orientační běh,), dominance šachovnice minimálním počtem figur, knihovna práce s polynomy.

Mráz (pozn. kráceno)

Kalkulačka s dlouhými čísly, kalkulačka s maticami, kalkulačka so zlomky, hra míny, žravá dáma, program generující vetvy magické štvorce zadaného rádu, piskvorky na ploše 19x19 políček, reverzi, kalkulačka s polynómy, řešení 8-smeriek.

Štanclová

Kalkulačka pracující s dlouhými čísly, maticová kalkulačka, vyhodnocování výrazu (v poli), názvy anorganických sloučenin, makroprocesor, zjednodušená verze regulárních výrazů, míny (minesweeper) v textovém modu, řešení malovaných krizovek, life, sudoku.

Malenko (pozn. kráceno)

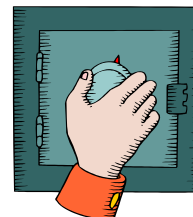
Hry - šachy, dáma, go, piskvorky, člověče nezlob se, (matfyzácké) pexeso. Had - had jezdí po obrazovce. Algoritmy nad maticemi - určitě GEM, inverzní matice. Algoritmy nad grafy: hledání nejkratší cesty podle více kritérií, hledání minimální kostry, topologické třídění. Vyhodnocování výrazů - vyhodnotí výraz v lineárním čase, možnost definice vlastních operátorů (s prioritou vyhodnocování) a funkcí. Komprese/šifrování - zkomprimujte/zašifrujte soubor buď vlastním (a chytrým) nebo jiným standardním algoritmem. Program, který se bude (bez znalosti hesla) pokoušet najít nezašifrovaný obsah.

U témat, kde je **poznámka kráceno** v případě zájmu poskytnu podrobnější popis. **Nejpozději do Vánoc** jsou studenti povinni mít **schválenou specifikaci zápočtového** programu. Specifikací se myslí krátký text (jeden až dva odstavce), který jednoznačně popisuje, o čem zápočtový program bude (například: hra dáma, atd... + další bližší popis), jaké bude mít funkce a jejich bližší popis (například: program bude umět hrát proti uživateli na určité úrovni, atd... + další bližší popis) a jakým způsobem bude zpracován a bližší popis tohoto zpracování (například: program bude mít grafické rozhraní, atd... + další bližší popis).

Jakékoli **dotazy** ke cvičení lze posílat na uvedenou e-mailovou adresu. Osobní konzultace ke cvičení lze dohodnout e-mailem (alespoň den předem).

1. Generování základních kombinatorických „objektů“. Napište program v Pascalu, který vygeneruje, tj. třeba vypíše, všechny:

- | | | |
|--------------------------------------|---|----------------------|
| (a) k členné kombinace z N prvků | } | bez opakování |
| (b) k členné variace z N prvků | | |
| (c) permutace N prvků | | |
| (d) k členné kombinace z N prvků | } | s opakováním |
| (e) k členné variace z N prvků | | |
| (f) permutace N prvků | | |



Předpokládejte, že vstupem programu jsou hodnoty k a N . Uvědomme si, že k členné kombinace z N prvků jsou vlastně k prvkové podmnožiny množiny, která má N prvků.

Řešení. Zkusme zatím zjednodušenou variantu. Budeme generovat všechny tříprvkové podmnožiny množiny, která obsahuje N prvků (například celých čísel).

```

program Vsechny_trojice;

const N = 10; { velikost množiny }
var y: array[1..N] of integer; { množina }

var i,j,k: integer; { indexy:
                        i - index prvního prvku trojice
                        j - index druhého prvku trojice
                        k - index třetího prvku trojice }

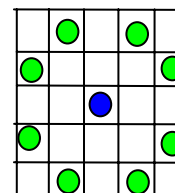
begin
  for i:=1 to N-2 do begin
    for j:=i+1 to N-1 do begin
      for k:=j+1 to N do begin
        writeln(y[i], ' ', y[j], ' ', y[k]);
      end;
    end;
  end;
end.

```

Zbytek za DCV.



2. Koník II. Je dána šachovnice o velikosti $n \times n$. Napište v Pascalu program, který nalezne posloupnost tahů koněm takovou, že každé políčko šachovnice kůň navštíví právě jedenkrát. Připomenutí dovolených tahů (modré kolečko značí výchozí pozici, zelená kolečka označují dovolené tahy):



Řešení. Napišme si řešící program nejprve česky (nebo anglicky, podle toho, co je komu bližší). Poté se tento slovní popis pokusíme transformovat v pascalský program.

Tahy budeme začínat z určité pozice. Máme tedy úlohu najít posloupnost tahů koněm, kterou navštívíme všechna políčka právě jedenkrát. Provedeme tah. Získáváme úlohu stejného typu pouze hledáme posloupnost o jeden tah kratší a jedno políčko šachovnice už je zaplněné. Hledání posloupnosti tahů s danými vlastnostmi délky jedna, tj. když už je šachovnice takřka vyplněná, vyřešit umíme. Dostáváme tedy jakýsi návod, jak úlohu řešit rekurzivně. Z dané pozice zkusíme provést tah a rekurzivně se pokusíme vyřešit úlohu velikosti o jedna menší. Když to jde, máme řešení (řešením je řešení menší úlohy, ke které přidáme provedený tah). Když menší úloha nemá řešení, zrušíme tah a provedeme jiný z možných tahů koněm. A opět řešíme rekurzivně úlohu o jedna menší. Jestliže se úloha nepovede vyřešit ani pro jeden možný počáteční tah koněm, pak nemá řešení. Strukturujme nyní tento popis. Předpokládejme, že někde stranou máme šachovnici, do které na jednotlivá políčka zapisujeme čísla tahů (detaily viz. cvičení).

*návod na vyřešení úlohy, když začínáme z políčka **A***

- 1. když je políčko **A** poslední nenavštívené, máme řešení** (je jím vyplněná šachovnice), jinak pokračujeme krokem **2**.
- 2. pro všechny možné tahy **t** koněm z políčka **A** provedeme následující**
- 3. zjistíme, zda je tah **t** možný, tj. výsledné políčko **A+t** ještě nebylo navštíveno**
- 4. když ne, pokračujeme zkoušením dalšího tahu**
- 5. když ano, pokračujeme rekurzivním řešením menší úlohy, tj. zapíšeme do šachovnice tah **t** (na místě **A+t**) a znovu použijeme tento návod, ale tak, že začínáme z místa **A+t****
- 6. když má menší úloha řešení, máme řešení** (je jím vyplněná šachovnice)
- 7. když menší úloha nemá řešení, smažeme z šachovnice tah **t** (na místě **A+t**) a pokračujeme zkoušením dalšího tahu**
- 8. když už jsou všechny tahy **t** vyzkoušené a přesto se úlohu nepodařilo vyřešit, úloha nemá řešení**

Zbývá přepsat tento popis do Pascalu. To je ale jednoduché, skoro mechanické. Bylo za DCV.

