

Pavel Töpfer  
Středoevropská olympiáda v informatice

*Pokroky matematiky, fyziky a astronomie*, Vol. 44 (1999), No. 4, 334--337

Persistent URL: <http://dml.cz/dmlcz/141011>

## Terms of use:

© Jednota českých matematiků a fyziků, 1999

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://project.dml.cz>

- [30] SÁNCHEZ-RUIZ, J.: *Maassen-Uffink Entropic Uncertainty Relation for Angular Momentum Observables*. Phys. Letts. A 181 (1993), 193–199.
- [31] SÁNCHEZ-RUIZ, J.: *Position-momentum entropic uncertainty relation and complementary in single-slit and double-slit experiments*. Phys. Rev. A 57 (1998), 1519.
- [32] SHANNON, C. E.: *A Mathematical Theory of Communication*. Bell Syst. Techn. J. 27 (1948), 629–653.
- [33] SRINIVAS, M. D.: *Entropic Formulations of Uncertainty Relations*. Pramana (Indian J. Phys.) 25 (1985), 369–377, a 24 (1985), 673.
- [34] UFFINK, J. B. M., HILGEVOORD, J.: *Uncertainty Principle and Uncertainty Relations*. Found. Phys. 15 (1985), 925–944.
- [35] VALLÉE, M. R.: *Aspects informationnels de certaines relations d'incertitude*. C. R. Acad. Sc. Parris 233 (1951), 1580–1581.
- [36] YANES, R. J., ASSCHE, W. VAN, DEHESA, J. D.: *Position and Momentum Information Entropies of the D-dimensional Harmonic Oscillator and Hydrogen Atom*. Phys. Rev. A 50 (1994), 3065–3067, a J. Math. Phys. 35 (1994), 4423.

## Středoevropská olympiáda v informatice

*Pavel Töpfer, Praha*

Studenti středních škol zájímaví se hlouběji o informatiku a programování dostávají již řadu let příležitost změřit své síly a schopnosti nejen v národních programátorských soutěžích, ale také v mezinárodním měřítku. Vedle celosvětové mezinárodní olympiády v informatice postupně vzniklo i několik obdobných regionálních soutěží. Studenti z České republiky se od samého počátku účastní středoevropské olympiády, jejichž prvních pět ročníků se konalo v Rumunsku (Cluj, 1994), v Maďarsku (Szeged, 1995), na Slovensku (Bratislava, 1996), v Polsku (Nowy Sącz, 1997) a v Chorvatsku (Zadar, 1998).

V letošním roce byla uspořádáním v pořadí již šesté Středoevropské olympiády v informatice pověřena Česká republika. Zajištění celé akce se ujali společně pracovníci Fakulty informatiky Masarykovy univerzity v Brně a Matematicko-fyzikální fakulty Univerzity Karlovy v Praze. Soutěž se konala ve dnech 2.–9. 9. 1999 v Brně v prostorách tamní fakulty informatiky. Skupina pracovníků brněnské fakulty v čele s doc. RNDr. Václavem Sedláčkem, CSc., se postarala o organizační a technické zabezpečení soutěže. Při zajišťování doprovodného programu jim pomohli také kolegové z gymnázia na tř. Kpt. Jaroše v Brně. Pracovní tým pražské Matematicko-fyzikální fakulty UK vedený doc. RNDr. Pavlem Töpferem, CSc., měl na starosti odbornou náplň — přípravu soutěžních úloh, testovacích dat, vyhodnocovacích programů i zajištění samotné soutěže a vyhodnocování a zpracování výsledků.

---

Doc. RNDr. PAVEL TÖPFER, CSc. (1960), Kabinet software a výuky informatiky MFF UK, Malostranské náměstí 25, 118 00 Praha 1

Celá olympiáda byla perfektně připravena a proběhla bez nejmenších závad. Také úlohy se ukázaly být vhodně zvolené, výsledky dosažené jednotlivými soutěžícími se téměř rovnoměrně rozložily do celé bodové škály od nuly až po plný počet 200 bodů. Čtyřčlenné soutěžní týmy reprezentující jednotlivé zúčastněné země byly vybrány na základě výsledků vrcholných národních soutěží v programování (u nás kategorie P matematické olympiády). My jsme jako pořadatelé využili možnosti postavit reprezentační týmy dva. Vedle 48 řádných soutěžících pracujících na počítačích na místě v Brně se dalších 52 zájemců zúčastnilo soutěže po Internetu. Do té se mohl přihlásit každý, kdo chtěl. Zadání úloh obdržel prostřednictvím Internetu zároveň se zahájením soutěže v Brně a stejnou cestou a ve stejném časovém limitu odevzdal své výsledky k vyhodnocení.

Vlastní soutěž probíhala ve dvou dnech. V každém byly zadány k řešení tři úlohy, za jejichž úplné vyřešení bylo možné získat dohromady 100 bodů. Plného zisku 200 bodů dosáhl jediný účastník, nadpoloviční většinu bodů (tj. více než 100) získalo 28 ze 48 zúčastněných studentů, mezi nimi i 7 z 8 českých reprezentantů. Kvalita odevzdaných řešení jednotlivých úloh se hodnotila pomocí předem připravené sady vstupních dat. Při testování šlo nejen o dosažení správných výsledků, ale také o rychlost výpočtu. Sledováním doby výpočtu se odlišily programy založené na lepším a na horším algoritmu a tento rozdíl se také odrazil ve výsledném bodovém hodnocení. Teoreticky správné, ale neefektivní programy získaly sice část bodů za úspěšný výpočet při zadání malých testovacích dat, neuspěly však při použití rozsáhlých dat pro překročení časového limitu stanoveného na dobu výpočtu.

Pravidla soutěže stanoví, že polovina soutěžících je oceněna medailí, přičemž zlaté, stříbrné a bronzové medaile se udělují v poměru 1 : 2 : 3. Celkem byly proto uděleny 4 zlaté, 8 stříbrných a 12 bronzových medailí. Z našich studentů získali Pavel Charvát a Pavel Nejedlý stříbrnou medaili, Jakub Bystroň, Zdeněk Dvořák a Jiří Svoboda medaile bronzové.

Jako ukázkou soutěžních úloh zadaných k řešení na 6. střeoevropské olympiádě v informatice jsme vybrali po jedné úloze z každého soutěžního dne.

## Hra s paritami

Se svým přítelem občas hrajete následující hru. Váš přítel napíše posloupnost nul a jedniček. Vy si vyberete souvislou podposloupnost (například úsek od třetího do pátého čísla včetně) a zeptáte se, zda tento úsek obsahuje sudý nebo lichý počet jedniček. Přítel odpoví a vy se ho stejným způsobem zeptáte na další podposloupnost. Vaším úkolem v této hře je určit celou posloupnost čísel.

Podezříváte svého přítele, že vám při hře ne vždy odpovídá pravdivě, a chcete mu lež dokázat. Proto jste se rozhodli napsat program, který vám v tom pomůže. Program dostane řadu vašich otázek a odpovídá, které jste na jednotlivé otázky od svého přítele obdrželi. Úkolem programu je určit, až do které odpovědi existuje nějaká posloupnost nul a jedniček vyhovující všem paritním podmínkám (přítel tedy může odpovídat pravdivě) a při které odpovědi je poprvé jisté, že přítel odpovídal nepravdivě.

### Vstup:

První řádek vstupního souboru `PARITY.IN` obsahuje jediné číslo udávající délku posloupnosti nul a jedniček. Tato délka nepřekročí 1 000 000 000. Na druhém řádku je zapsáno číslo určující počet položených otázek a odpovědí. Otázek nebude nikdy více než 5000. Každý z dalších řádků popisuje jednu otázku a odpověď. Obsahuje dvě celá čísla (pozici prvního a posledního čísla v dotazovaném úseku) a jedno slovo „even“ nebo „odd“ (odpověď na otázku, tzn. paritu počtu jedniček v úseku, přičemž „even“ znamená sudý počet jedniček, zatímco „odd“ lichý).

### Výstup:

Výstupní soubor `PARITY.OUT` je tvořen jediným řádkem obsahujícím jedno číslo. Toto číslo  $X$  udává, že existuje posloupnost nul a jedniček vyhovující prvním  $X$  paritním podmínkám, ale žádná posloupnost nevyhovuje prvním  $X + 1$  podmínkám. Pokud existuje posloupnost nul a jedniček vyhovující všem zadaným podmínkám, bude toto číslo  $X$  rovno počtu všech položených otázek.

#### Příklad 1:

PARITY.IN	PARITY.OUT
10	3
5	
1 2 even	
3 4 odd	
5 6 even	
1 6 even	
7 10 odd	

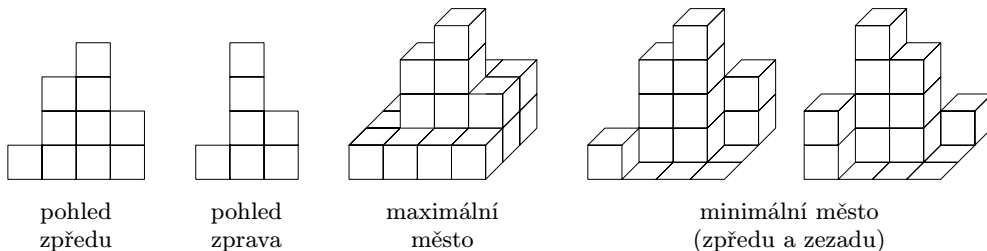
#### Příklad 2:

PARITY.IN	PARITY.OUT
10	5
5	
1 2 even	
1 4 even	
2 4 odd	
1 10 even	
3 10 even	

## Město z kostek

Děti si rády hrají s dřevěnými kostkami. Většinou z nich staví vysoké věže, ale malý Honzík má jiné plány. Chce z kostek vystavět veliké město. Jeho tatínek mu koupil obdélníkový stůl, na který se vejde na šířku přesně  $K$  kostek a na délku  $L$  kostek. Honzík se zamyslel a rozhodl se vytvořit nejprve plán města, dříve než se pustí do vlastní stavby. Na stůl si namaloval čtverečkovou síť velikosti  $K \times L$  čtverců. Na některé z těchto čtverců namalované sítě chce pokládat věže sestavené z jedné nebo více kostek, zbylé čtverce zůstanou prázdné. Protože stůl je obrovský, Honzík nechce přesně plánovat pro každý čtverec, kolik kostek na něj postaví na sebe. Chce se jen předem rozhodnout, jak bude jeho město vypadat při pohledu zředu a zprava. Tyto dva pohledy (dvourozměrné projekce plánovaného města) si Honza namaloval na papír. Na obrázku vidíte příklad Honzíkových nákresů a jim odpovídajícího města z kostek.

Honzíkův otec má obavy, zda budou Honzíkovi na plánovanou stavbu stačit jeho kostky. Žádá vás proto o napsání programu, který spočítá minimální a maximální počet kostek, z nichž lze postavit město podle Honzíkových nákresů. Program má také určit, zda je vůbec možné postavit nějaké město odpovídající těmto nákresům.



*Vstup:*

První řádek vstupního souboru `TOWN.IN` obsahuje dvě celá čísla  $K, L$  — šířku a délku stolu (u dané počtem kostek). Stůl není širší ani delší než 100 000 kostek. Následující řádky vstupního souboru obsahují popis předního pohledu na město. Popis je tvořen výškami viditelných staveb po jednotlivých čtvercích zleva doprava (výšky se opět udávají počtem kostek). Na každém řádku je zapsáno jediné číslo, takže počet řádků s popisem pohledu na město zředu je přesně roven šířce stolu  $K$ . Další  $L$  řádků vstupního souboru obsahuje podobným způsobem uložený pohled na město z pravé strany. Výšky jednotlivých sloupců kostek jsou tentokrát uloženy v pořadí odředu dozadu. Můžete předpokládat, že žádná stavba ve městě nebude vyšší než 5000 kostek. Maximální počet kostek potřebný k postavení města nepřekročí hodnotu 2 000 000 000.

*Výstup:*

Výstupní soubor `TOWN.OUT` obsahuje jediný řádek. Pokud není možné postavit žádné město podle zadaných pohledů zředu a zprava, bude soubor obsahovat pouze text „No solution.“. V opačném případě jsou ve výstupním souboru zapsána dvě celá čísla oddělená mezerou. První z nich udává minimální a druhé maximální počet kostek, které může Honzík použít pro stavbu města podle svých nákresů.

*Příklad 1:*

```
TOWN.IN      TOWN.OUT
4 3          10 21
1
3
4
2
1
4
2
```

*Příklad 2:*

```
TOWN.IN      TOWN.OUT
2 2          No solution.
4
1
1
3
```