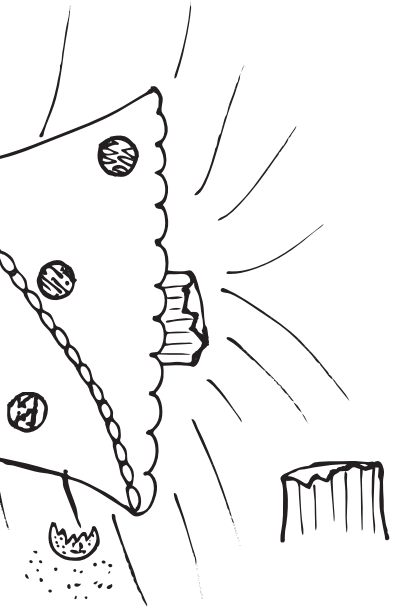
azují řetězce za jiné, co dělat, když regex pro náš
ný řetězec prostě neexistuje a jak s tím vším sou-
k Norris.



Na případné dotazy vám rádi odpovíme také na adrese *ksp@mff.cuni.cz* a v diskusním fóru na našem webu.

Třetí série třiadvacátého ročníku KSP

Edsger Dijkstra byl slavný holandský myslitel. Byl to samotářský, konzervativní člověk, kterému se jen velmi těžko určuje obor, jímž se zabýval. Narodil se roku 1930 v Rotterdamu, kde zůstal až do svých vysokoškolských studií teoretické fyziky. Později žil a učil v Eindhovenu, kde se věnoval praktické i teoretické matematice a informatice.

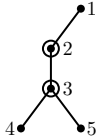
Zabýval se mimo jiné i grafy. Jedním z nejznámějších algoritmů, které vymyslel, je hledání nejkratší cesty v ohodnoceném grafu, jenž po něm nese jméno a můžete si ho přečíst třeba v naší kuchařce.[1] *My bychom po vás teď chtěli vymyslet jeden trochu jednodušší, ale zdánlivě podobný algoritmus.*

23-3-1 Úsporný kořen — 9 bodů

Na vstupu dostanete neohodnocený strom[2] zadaný počtem vrcholů a seznamem hran. Vrchol s hloubkou x bude takový vrchol, od kterého je každý jiný vrchol vzdálený maximálně x. Úsporný kořen je vrchol s nejmenší možnou hloubkou. Nalezněte všechny úsporné kořeny stromu.

Příklad vstupu:

```
5
1 2
2 3
3 4
3 5
```



Výstup: `2 3`

Jako správný samotář bydlel na vesnici, takže do školy jezdil jen v úterý. Vedl tam i seminář, kterému se příhodně říkalo Tuesday Afternoon Club, kde se řešily úlohy podobné těm z KSP. Kdykoliv tam či jinde studenti příliš hlasitě šuškali (to asi dobře znáte), nekřičel, ale naopak začal šeptat. To mělo ohromný efekt – všichni ztichli. Takový měl respekt.

Když se později stěhoval do Ameriky a cesta mu připadala dlouhá, vymyslel úlohu, kterou pak zadal americkým studentům při jejich prvním semináři Tuesday Afternoon Club.

23-3-2 Nejkratší cesta přes oceán — 14 bodů

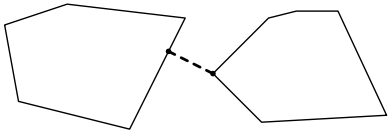
⚠ Pro zjednodušení si severní Ameriku i Evropu představme jako dva konvexní n-úhelníky, které se neprotínají. Chcete nalézt nejkratší cestu mezi nimi. Na vstupu nejdříve dostanete souřadnice Ameriky a potom souřadnice Evropy jako vrcholy v pořadí, v jakém leží na obvodu.

Vaším úkolem je najít takové dva body, jeden na obvodu Ameriky a druhý na obvodu Evropy, aby jejich vzdálenost byla co nejmenší. Pokud je víc možností, stačí vypsat libovolnou z nich.

Vstup (znázorněný obrázkem):

```
5
0 75
10 20
90 0
130 80
45 90

6
185 5
275 10
240 85
210 85
190 80
150 40
```



Výstup (čárkovaně):

```
118 56
150 40
```

Krom tisíce jiných věcí přemýšlel, jak počítače naučit počítat, třeba odpovědět na zadání `1+2*3`. *K tomu účelu znovu objevil postfixový zápis a objevil, jak na něj běžný infixový zápis rychle převést. Pokud to chcete umět taky, můžete se podívat třeba do Wikipedie.*[3]

Abychom nezůstali jen u teorie, podílel se na vývoji programovacího jazyka ALGOL 60 a později i jeho prvního překladače. Tento jazyk vznikl pro snadný zápis algoritmů a jako konkurence tehdy mohutně nasazovaného BASICu.

Edsger Dijkstra vůbec velmi brojil proti příkazu `goto` *a zasazoval se o strukturované programování. Příkaz* `goto` *po-*

[1] `http://ksp.mff.cuni.cz/tasks/20/cook4.html`
[2] `http://ksp.mff.cuni.cz/tasks/21/cook3.html`
[3] `http://cs.wikipedia.org/wiki/Shunting-yard_%28algoritmus%29`

rem lze psát víc navazujících linií jedním tahem, o, musí se pero zvednout a začít jinde. Kolikrát je potřeba pero zvednout?

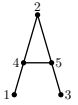
u dostanete neorientovaný graf o N vrcholech a ch a vypište, kolika nejméně tahy lze nakreslit.

mě šetříme, takže je zakázáno jakoukoli hranu na- ce než jednou. Jinak řečeno, nesmíte se vracet po slených liniích.

stupu:

Jiný příklad:

```
5 5
1 4
4 2
2 5
5 3
4 5
```



výstup: 2

na je praktická a řeší se ve vyhodnocovacím sys- dEx.[4] Formát vstupu a výstupu, povolené jazyky echnické informace jsou uvedeny v CodExu přímo

z mnoha jeho textů (EWD 1250) – známý problém ů, lodky a řeky. K řece přišly dva páry a potřebují řeku tak, aby nikdy nezůstal sám pán z jednoho mou z druhého páru. Lodka unese maximálně dva

ouples, the river, and the little boat

re is the problem:

husbands and two wives have to cross er in a boat which can hold only two e. How can they cross so that no woman the company of a man unless her hus- is also present? "

(Copyright © 1967 by Morris Kline)

Analogii můžeme najít v hudbě, kde se takhle označují díla slavných skladatelů, asi nejznámější jsou BWV (Bach-Werke-Verzeichnis) a HWV (Händel-Werke-Verzeichnis). Zásadní rozdíl je ovšem, že Dijkstra si to čísloval sám, kdežto hudebníci to neřešili a udělal to za ně někdo jiný o mnoho let později.

EWD shromažďuje pan Ham Richards. Jednou, když je nesl, zakopl a rozsypaly se mu po podlaze.

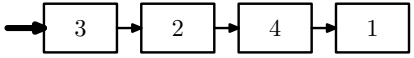
23-3-5 Rozházené EWD — 7 bodů

Chudák pan Richards má jen svou zapomětlivou hlavu a pár papírů, tak budete muset vymyslet, jak setřídit EWD v konstantní paměti. To jest, že si může udělat třeba 1000 záznamů, ale ne pro každou z N EWD jeden. Vámi spotřebovaná paměť prostě na N vůbec nesmí záviset (a N může být libovolně velké – argument, že EWD je konečně mnoho, vám neprojde). Dávejte si pozor na rekurzi, spotřebovává tolik paměti, jak hluboko je zanořená.

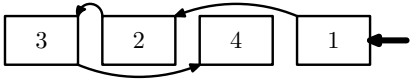
Přeházená EWD budeme reprezentovat jako spojový seznam. V programu dostanete ukazatel na první prvek spojového seznamu, kde je číslo EWD a ukazatel na další. Vaším úkolem je ho setřídit a vrátit ukazatel na první prvek (nejstarší EWD).

Spojový seznam už máte v paměti, vaším úkolem je přepojit jej do setříděného stavu.

Příklad před setříděním:



A po setřídění:

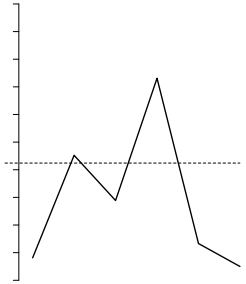


Z úplně jiného soudku je jeho návrh operačního systému THE multiprogramming system, zaměřeného na sekvenční zpracování úloh a s podporou multitaskingu a paralelizace. Velkého rozšíření se sice nedočkal, nicméně myšlenky vytvořené pro něj se ujaly. Jestli vás paralelní svět zajímá, měli jsme o něm kdysi seriál[6] a také o něm byla série úloh před pár lety v Matematické olympiádě.

[5] http://www.cs.utexas.edu/users/EWD/welcome.html
[6] http://ksp.mff.cuni.cz/tasks/12/

Příklad vstupu:

```
6 5
8.124 45.223 28.8723 73.117 13.3 5.0
```



Výstup (vyznačen čárkovanou čarou): `42.42` (4×)

23-3-7 Automaty stokrát jinak — 12 bodů

Tento text navazuje na předchozí dvě série, některé pasáže nemusí být lehce pochopitelné bez jejich znalosti.

Minule jsme si popsali nedeterministický konečný automat (NKA) s ε-přechody. Definovali jsme, že vstup přijme, pokud v něm existuje alespoň jedna možnost, jak při čtení onoho vstupu skončit ve výstupním stavu.

Běžný program ale neumožňuje paralelně zkoumat všechny větve, kterými by mohl procházet. To bychom si museli pořídit třeba paralelizátor jako v jednom ze starých ročníků olympiády.

Mohli bychom však zkusit převést NKA na DKA, který programem simulovat velmi jednoduše umíme. Dá se dokázat, že to jde vždycky (i když výsledný automat může mít až exponenciální velikost), ale obvykle se to dělá ukázáním obecného postupu, takže si to ukážeme až v řešení.

Úkol 1 [6b]: Vymyslete, jak simulovat NKA a jak převést NKA na DKA. Pokud vám to nepůjde ve vší obecnosti, máte šanci získat 2 body za převod tohoto konkrétního automatu: