

Korespondenční Seminář z Programování

ZAČÁTEČNICKÁ KATEGORIE

27. ročník

KSP-Z

Březen 2015

Už třetí série začátečnické kategorie KSP je u konce a jako obvykle vám přineseme autorská řešení úloh. Věříme, že vám pomohou k tomu, abyste se v programování a hlavně v řešení problémů pořádně zlepšovali. Zkusíte se podívat, které úlohy byste řešili jinak a které zase stejně jako autoři nebo jak by se daly řešit úlohy, se kterými jste si nevěděli rady?

Gratulujeme všem, kdo získali nějaké body! A jako obvykle se nás nebojte zeptat, pokud vám cokoli není úplně jasné. Obrátit se na nás můžete přes fórum na našich stránkách nebo e-mailem ksp@mff.cuni.cz.



Řešení třetí série začátečnické kategorie 27. ročníku KSP

27-Z3-1 Kevin nabíječ, s.r.o.

Když jste nad úlohou chvíli přemýšleli, připadne si zkusíte ryzna zapojení, asi jste přišli na to, že na zpusobu zapojení výřeb nezalazi, dokud budou všechny produhovavacky prijpojeny k té jedné ve zdi. Každá produhovavka jednu zdirku pouzije pro své napájení a k jich poskytnu pro libovolné další použít. Pokud použijeme všechny dostupné produhovavacky, můžeme prostě spočítat součet všech $k-1$. To uděláme nejsnáze tak, že sečteme všechna k a na konci odečteme N (a přidáme jedničku za závorku ve zdi).

V zadání byl ale malý háček, který jsme se snažili naznačit obrázkem. Kevin si nakoupil i produhovavacky, které měly mla zdířek, a takové zapojovat nedujeme. Takže je stačí prostě z řešení vyloučit. Tento chyták byl jen v posledním testu, takže i pokud jste si ho nevšimli, mohli jste dostat většinu bodů. Na takové chytáčky si ale dávejte pozor.

Program (Python 3):

<http://ksp.mff.cuni.cz/viz/27-23-1.py>

Oudru Hlavatý

27-Z3-2 Nedějí vliagen

Maximální délka slova na vstupu byla v zadání 100. To je dostatečně malá konstanta na to, abychom se jí nemuseli zabývat. Soustředíme se hlavně na to, aby náš program byl co nejrychlejší vzhledem k počtu slov na vstupu.

Připravíme si spojový seznam všech zadáných slov a druhý spojový seznam, ve kterém budou také všechna slova, akorát napápná pozpátku. Tedy stačí jen najít slova, která jsou v obou seznamech.

To se dá dělat více způsoby. Nejsnazší je prostě každé slovo z prvního seznamu porovnat se všemi ze druhého a zapast si ho stranou, pokud se s nějakým shoduje. To by mělo časovou složitost $O(N^2)$, kde N je počet slov na vstupu.

O něco mazanější a výrazně rychlejší způsob je oba seznamy setřídít abecedně a potom je šikovně zkoumat najednou. Na třídění poslovnosti je spousta algoritmů, které skončí v čase $O(N \log N)$.¹

Jak přesně tedy budeme seznamy prohlížet? Budeme postupně procházet oba seznamy najednou a v každém kroku se podíváme na první slova v obou seznamech. Pokud jsou stejná, nalezení jsme požadovanou shodu. Jinak z nich vybereme to lexikograficky menší (neboli to, které by se ve

	řešitel	škola	ročník	série	celkem	
31.-32.	Jan Burda	G-Holice	0	6	33,0	69,5
	David Ucháč	eduSOS PA	2	2	51,5	69,5
33.	Jakub Nenda	GTTNovakBO	4	3	0,0	69,0
34.	Pavel Turninský	G Brandýs	2	2	28,0	63,8
35.-36.	David Beláček	GSOS Podb	3	2	18,0	57,0
	Zuzana Šimečková	GČeskaČeč	4	3	29,0	57,0
37.	Daniel Plíškal	G BO-Reč	1	2	0,0	56,0
38.	Zuzana Drazdová	GČeskaČB	4	2	20,0	52,0
39.	Ondřej Svanda	G BO-Reč	4	2	0,0	50,0
40.	Václav Fabík	ZŠKřidloBO	4	2	8,0	48,0
41.	Josef Vávra	SJae	6	2	27,4	47,4
42.-43.	Petr Klanica	GJarošBO	2	1	0,0	45,0
	Zoltán Onódy	SPSE NZám	3	1	0,0	45,0
44.	Ondřej Mleotota	SPMasarLI	4	2	7,0	42,8
45.-50.	Patrick Bak	G Solrance	4	1	0,0	40,0
	Tat Dat Duong	G Wicht	2	1	0,0	40,0
	Jakub Lukáš	GNAlejPH	2	2	0,0	40,0
	Jiří Sejkora	GvodetrahPH	3	1	0,0	40,0
	Jan Soukup	GKlatovy	4	1	0,0	40,0
	Jan Václavěk	GJUnOHl	4	1	0,0	40,0
51.-52.	Janek Havatý	ZŠ DuhelCB	-4	2	6,0	39,0
	Michaela Štolová	G,Sokolov	3	2	28,0	39,0
53.	David Tvrdý	GHeyrovPH	2	1	0,0	38,0
54.	Jan Vozar	G UherBrod	1	6	18,0	37,5
55.	Jonáš Přibyl	ZŠ ČJBrod	0	1	36,0	36,0
56.	Michaela Svatošová	GKopelaraPH	1	3	13,0	35,0
57.-59.	Lukáš Holecý	GJlep	3	2	0,0	28,0
	Ivana Krumlová	GJarošBO	2	3	0,0	28,0
	Benedikt Žour	G UherBrod	0	5	0,0	28,0
60.-61.	Michael Kozel	GZborovPH	1	3	6,0	24,0
	Roman Ondráček	GBoškovice	1	3	6,0	24,0
62.	Dominika Tangelová	G,Nymburk	2	1	0,0	22,0
63.-64.	Jan Šlacky	G Benesov	4	1	0,0	21,0
	Andrej Černák	G JF Šaľa	1	2	12,0	21,0
	Victoria Maria Nájares Romero	GZborovPH	1	1	0,0	20,0
65.	Miroslav Břežák	GLesmiZim	0	1	0,0	18,0
66.-70.	Michal Nekvinda	BiGYBBHK	4	1	0,0	18,0
	Alexej Popović	SlovnaGOL	3	1	0,0	18,0
	Vojta Stanek	PORGPpa	4	1	0,0	18,0
	Petr Šima	GKlatovy	1	5	0,0	18,0
71.	Martin Zima	G,Holice	1	1	16,0	16,0
72.	Vojtěch Pejša	G,Koim	4	1	0,0	15,0
73.	Milan Kutubala	GTAjBarBys	3	1	0,0	14,0
74.	Filip Priečinský	SG Zilina	2	2	7,0	13,5
75.-81.	Michaela Bačová	G UherBrod	4	3	0,0	10,0
	Antonin Brůstík	G UherBrod	4	5	0,0	10,0
	David Dvořáček	G UherBrod	4	3	0,0	10,0
	Liboš Kolumber	SpojŠ Popr	3	3	7,0	10,0
	Viktor Kovařík	G UherBrod	4	4	0,0	10,0
	Markéta Machalová	G Wicht	2	2	2,0	10,0
	Jakub Smalčovský	G Pezinok	3	1	10,0	10,0
82.	Matěj Hudec	GirkG Pizaň	3	1	9,4	9,4
83.-86.	Matěj Hočkirco	TAPoprad	1	1	0,0	8,0
	Ján Pavlus	GTTNovakBO	2	1	0,0	8,0
	Martin Sklenář	GTAjBarBys	3	1	0,0	8,0
	Petr Zelina	GJarošBO	2	1	0,0	8,0
87.	Vít Gadurek	Neuvaděná	0	2	4,0	5,0

Chcete-li s námi komunikovat sítrovanné a bezpečně, můžete si otevřít náš HTTPS certifikát – jeho SHA1 fingerprint je: **0E: D9: B6: E5: 6F: B0: 51: D9: 66: EB: E9: 29: E4: 58: AB: 5F: 99: D6: FD: A3.**

27-Z3-3 Superstromy

Podle počtu správných řešení se zdá, že jste s úlohou neměli příliš problémů. Ono také nebylo potřeba vymýšlet nic světoborného. Stačilo si jen uvědomit, že si nikdy neuskódníme tím, že stromy rostoucí pomažíme jako první.

Proč to platí? Vždy alespoň jeden strom doroste jako poslední. Jak bychom mohli naše řešení zlepšit? Jediné tím, že tento poslední strom zasaďme v nějaký dřívější den. Pokud jsme však v dřívějším dnu sazili pouze pomalejší stromy, prohozením si pářky akorát odložíme na později.

Optimální pořadí stromů tedy dostaneme tak, že si je seřadíme sestupně dle počtu dnů, které rostou. Následně jen stačí najít maximum ze součtu dnů v čísla dne, ve který daný strom zasaďme.

Dostali jsme tak řešení s časovou složitostí $O(N \log N)$ a pamětovou $O(N)$. To bohatě stačilo na plný počet bodů. Když si však všimneme nízkého limitu na dobu růstu, mohli řešení o něco vyřešit.

Program (Python 3):

<http://ksp.mff.cuni.cz/viz/27-23-3.py>

Protože existuje pouze $T = 1000$ různých typů stromů, můžeme si v poli velikosti T uložit, kolik kterých z nich máme.

Díky tomu zvádneme čísla třítří v čase $O(N+T)$. Dokonce si i vyznáme Jan s $O(T)$ paněti, protože si nemusíme pamatovat celý vstup.

Ačkoli popsaná myšlenka není vůbec složité, vyslovila si svůj vlastní názor. Takovému třídění čísel se říká *counting sort*. Vyplatí se jej použít v případech, kdy chceme serafit velké množství stejných hodnot.

Program – counting sort (Python 3): <http://ksp.mff.cuni.cz/viz/27-23-3-counting.py>

Jenka Hadram

27-23-4 Hobo Rally

V této úloze stačí vymyslet, jak v programu reprezentovat herní plochu a roboty a krok po kroku odsimulovat, co se děje.

Je mnoho možností, jak k problému přistoupit. Hlavní je nějakým způsobem reprezentovat robota, třeba jako objekt nebo klidně jako seznam čísel udávajících jeho pozici a orientaci. Například seznam $[7, 1, 2, D]$ by mohl reprezentovat robota číslo 7, který je v prvním řádku, druhém sloupci a dívá se směrem dolů.

Potom stačí vymyslet, kam roboty dát. Je rozumné mít je v poli indexovaném podle jejich identifikátorů, abychom se snadno dostali k robotovi, kterého zrovna potřebujeme.

Často navíc budeme zjišťovat, co je na nějakém políčku. Nejvhodnější způsob je procházet souřadnice všech robotů a zjišťovat, jestli se nějaká neshoduje se zkompaným políčkem. To by ale mohlo trvat dlouho a děláme to často. Pokud nám záleží na rychlosti, můžeme si udělat pomocné dvouzměrné pole, kde na každém políčku bude číslo robota, který na něm je (nebo rovnou odkaz na něj), nebo -1 (resp. nulový odkaz), pokud tam žádný není.

Tedy už nebudete muset načíst pozice a orientace robotů a umisřit je do mřížky datové struktury. Otáčení robotů je taky triviální. Pokud směry reprezentujeme písmeny, stačí k tomu několik podmínek. Pokud reprezentujeme smery čísly 0, 1, 2, 3, můžeme si pomocně vytvořit, která nám umožní zjišťovat vždy pole po dané čítrny. Tedy když se chceme otočit o tři doprava, tak k souřadnému směru přičteme trojku a výsledek vymodulíme čtyřmi.

Zajímavější je pohyb. Musíme si dát pozor, že je potřeba simulovat krok po kroku. Tedy posun o pět políček bude muset rozložit na pět kroků a každý vyhodnotit zvlášť. Pokud nám v nějakém kroku nestojí nic v cestě, tak je to snadné. Stačí přepsat posunutému robotovi souřadnice a připadně upravit pomocnou tabulku.

Pokud nám stojí něco v cestě, řekneme tomu, ať udělá krok stejným směrem a podle stejných pravidel jako my. Tedy pokud nám bude stát něco v cestě, tak tomu taky řekne, ať se to posune. Takto se o tom dozví celá řada. Je důležitě ošetřit případ, kdy se řada už nemá kam posunout (je před ní zed). Kvalitě tomu je potřeba ještě před samotným posunutem nejdříve ověřit, zda se vůbec můžeme pohybovat, a teprve potom posun provést.

Po provedení všech kroků stačí vypsat postupně souřadnice a orientace robotů.

Program (Python 3): <http://ksp.mff.cuni.cz/viz/27-23-4.py>

Martin Spaněl

27-23-5 Dřevěná slacklajna

Jak už všichni víme z hodně matematických trojúhelníků, je-li nějaká strana je delší než součet všech ostatních, nejde nakreslit. A stejně tak to platí i pro mnohoobelník. Když by Kevin se Sárou položili všechna prkna kromě největšího za sebe do jedné linie a tato linie by byla kratší než ono největší prkno, nemohou si okruh vůbec vytvořit.

Základem řešení bude tedy tato myšlenka. Délky prken si nejprve seřadíme, ideálně od největšího po nejmenší. Součet všech délek označíme S , délku největšího prkna D .

Pokud je:

- $S - D > D$, okruh lze postavit ze všech prken, jejichž délka je D nebo menší.
- $S - D \leq D$, tak největší prkno nelze použít. Od S odečteme D a celé pokračování provedeme znovu, akorát D již bude délka dalšího největšího prkna v pořadí.

Na konci stačí zkontrolovat, že nám zbyla alespoň tři prkna, ze kterých může Kevin se Sárou okruh postavit.

Časová složitost bude $O(N \log N)$, kde N je počet prken. Projítí seřazeného seznamu délky prken bude lineární, jelikož pro každé prkno provedeme pouze konstantně mnoho operací. Ovšem seřazení prken bude trvat $O(N \log N)$.

Program (Python 3): <http://ksp.mff.cuni.cz/viz/27-23-5.py>

Katka Záhorská

27-23-6 Red Bull dává křídla

Nedostali jsme příliš mnoho řešení, což nás mrzí. Pojdme si ukázat, že tato úloha nebyla tak těžká, jak se vám možná zdála.

Jelikož procházíme stavovým prostorem (což neznamená nic jiného, než že máme hrací plochu a na každém políčku můžeme být buď jako král, nebo jako kní), tak si musíme roznyslet, jak jej budeme reprezentovat.

Pro tuto konkrétní úlohu se nám zdá vhodné mít dvě dvou-rozměrná pole, která budou odpovídat velikosti hrací plochy, tedy velikosti $M \times N$.

Jedno pole bude pro místa, kam jsme došli jako král, a druhé bude pro místa, kam jsme došli jako kní.

Vždy z každého dosaženého políčka zkusíme jít do všech políček ve druhém poli, na která se daná figurka může dostat a ještě jsme tam nebyli. (Ve druhém poli, protože se taly konem a králem střídají.)

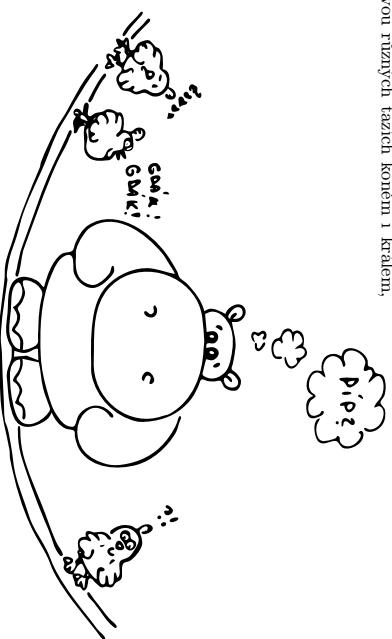
To, jestli jsme v některém stavu už byli, si budeme značit číslem udávajícím, v kolikátém kroku jsme tam došli. Což nám pomůže při debugování, neboť víme, kam nám to kdy skočilo. Ale taky podle toho dokážeme zrekonstruovat cestu.

Pro nalezení cesty pak stačí jít z cíle do startu po stavech, která mají právě o jedna menší číslo, než ve kterém jsme. Před výjsem tuto posoupit musíme ještě otočit. Rozmyslete si, že v tomto případě nelze udělat trik, že bychom hledali cestu z cíle do startu, a tím se vyhnuli otáčení poslopnosti.

Posledním krokem je si rozmyslet, jak rychle zjistit, kam jsme již skočili. Nejlepší bude si u pamatovat v nějaké struktuře, do které budeme umět v konstantním čase vkládat prvky na konce a vyjmout první prvek. Takové strukturu se říká fronta.

Nyní již jen zbyvá zamalyzovat, jak dobrý algoritmus jsme vymysleli. Projítí všech stavů nám bude trvat $O(N \cdot M)$, protože právě tolik je stavů a v každém stavu děláme konstantní množství operací, a stejně tolik paněti budeme potřebovat.

Závěrem ještě dodáme, že použít jen jedno společné pole pro kroky koněm i králem nestačí, protože na tom, jakou figurku jsme se do které pozice dostali, závisí i následné možné pohyby z tohoto políčka dále. A pokud políčka zvládneme dosáhnout ve dvou různých tazích koněm i králem,



Výsledková listina třetí série začátečnické kategorie 27. ročníku KSP

číslo	řezitel	škola	ročník	série	Z3-1	Z3-2	Z3-3	Z3-4	Z3-5	Z3-6	série	celkem
0.												
1.	Jakub Pele	G_UherBrod	1	7	8	10	10	12	12	14	66,0	198,8
2.	Jan Kalfer	GČesBrod	-1	3	8	10	10	12	12	10	62,0	177,8
3.	Martin Schentebren	G_MNám_Třb	3	3	8	10	10	12	12	11	63,0	184,0
4.	Jakub Matěna	GČesKolPřH	3	3	8	10	10	12	12	5	57,0	183,0
5.	Jiří Štěpanovský	G_MNám_Třb	3	3	8	10	10	12	12	4,5	56,5	169,0
6.	Lukáš Červený	G_Trutnov	1	3	8	10	10	12	5	8	57,0	166,0
7.	Vojtěch Lukáš	G_CeskaCB	3	3	8	10	10	12	9	9	53,0	162,5
8.	Lukáš Mířan	GČeskaCB	1	3	6	10	10	12	12	12	51,3	148,3
9.	Lukáš Vlček	G_MHradSPJL	1	3	8	10	10	12	10	1	51,0	145,0
10.	Jakub Jirňal	G_UngarnLT	0	3	8	10	10	12	9,5	4	53,5	140,5
11.	Miroslav Hrabal	G_TomavaOL	1	3	8	10	10	0			28,0	140,0
12.	Michal Topfer	G_DejPeKaMB	2	6	8	10	10	10			40,0	129,0
13.-14.	Karolina Kudryňová	G_MLerchaBO	4	3	8	10	10	10	12	12	40,0	120,0
	Jiří Moravčík	G_UHradisté	1	3	8	10	10	12			40,0	120,0
15.	Roman Beňo	G_HroncabaBA	2	2	8	10	10	10	12	6	58,0	113,0
16.	Teriza Kotěšová	G_Klatovy	4	3	8	10	10	12			40,0	110,8
17.	Lukáš Fhrnk	G_LesniZhm	2	3	8	10	10	12	11		51,0	100,0
18.-19.	Tomáš Tereň	G_TajBambšvs	3	2	8	10	10				39,0	99,0
	Tomáš Troján	G_Cheb	-1	3	8	0	10	10	12	11	30,0	99,0
20.-21.	Zdeněk Pavlák	G_MKrášáPřL	3	5	8	10	10				30,0	98,0
22.	Pavel Souček	G_Nymburk	1	2	8	10	10	12			40,0	98,5
23.	Jakub Rozlívek	G_OA Chodov	3	1	8	4	10	10	12	9	42,5	93,5
24.	David Zátěck	G_ZborovyPH	2	2	8	10	10	1			29,0	84,0
25.	Michal Převrtil	G_Klatovy	2	3	8	8	10				8,0	82,0
26.	Tomáš Chvořta	G_PřH	4	4	8	10	10	12			40,0	80,0
27.	Matúš Madar	G_HornMíchal	3	3	6	6		10		11,5	7,3	76,3
28.	Daniel Nigrm	G_UstanevPH	2	3	6	6			10		28,8	75,8
29.-30.	Nhat Minh Dinh Huy	G_Kataai	2	3	8	6	10	10		6	14,0	74,0
	Jan Mířaz	G_Holice	1	2	8	10	10	12	9	12	26,0	72,0
											61,0	72,0

Program (Python 3): <http://ksp.mff.cuni.cz/viz/27-23-6.py>

Vojta Sejkora