Korespondenční Seminář z Programování

ZAČÁTEČNICKÁ KATEGORIE

34. ročník **KSP-Z** **Únor 2021**

Řešení čtvrté série začátečnické kategorie 34. ročníku KSP

34-Z3-1 Otočená morseovka

Začneme tím, že si text převedeme do morseovky. Pro každé písmeno v textu známe jeho odpovídající kód a tyto kódy pro jednotlivá slova spojíme za sebe, kde mezi kódy pro jednotlivá písmena připíšeme `/`, jak bývá v morseovce zvykem. Tak dostaneme kód pro celou zprávu.

Nyní nás zajímá, zda je tento kód palindrom, tedy zda se čte zepředu stejně jako zezadu. Pokud tomu tak je, pak budeme-li zprávu číst zezadu, interpretujeme ji stejně jako popředu. Zjistit, zda je řetězec palindrom, můžeme například tak, že jej zkopírujeme, otočíme a následně porovnáme s původním řetězcem.

Jelikož počet znaků v morseovce je pro každý znak abecedy nejvýše 5, celková velikost zprávy v morseovce je $\mathcal{O}(L)$, kde L je délka původní zprávy. Otáčení a porovnávání řetězců je lineární vzhledem k délce řetězce, tedy časová složitost je pro každý příklad $\mathcal{O}(L)$ a celková složitost přes všechny zprávy je lineární se součtem délek zpráv. Jelikož můžeme zprávy zpracovávat postupně, celková paměťová složitost je lineární vzhledem k nejdelší zprávě.

Ukažme ještě, že převádět zprávy do morseovky vůbec není potřeba. Protože písmena oddělujeme `/`, můžeme palindrom kontrolovat po celých písmenách. Kód prvního písmene musí být otočením kódu posledního písmene, kód druhého písmene otočením předposledního atd. Tudíž si stačí pro každé písmeno zapamatovat, jaké písmeno odpovídá jeho obrácenému kódu. Například otočením kódu `B` dostaneme kód `V` a naopak. Otočenému kódu `C` žádný kód neodpovídá, pokud se tedy ve zprávě vyskytuje, zpráva určitě není palindrom.

Stačí tedy, když projdeme zprávu a pro i-té písmeno zepředu zkontrolujeme, že i-té písmeno zezadu je jeho protějšek. Časová i paměťová složitost zůstává stejná, ale s lepšími konstantami, protože zpráva zapsaná písmeny je kratší než zpráva v morseovce.

Program (Python 3):
`http://ksp.mff.cuni.cz/viz/34-Z3-1.py`

Úlohu připravili: Honza Černý, Kiki Prokopová, Jirka Sejkora

34-Z3-2 Tečkový displej

Začneme načtením všech zastávek a porouchaných displejů. Potom každý porouchaný displej porovnáme se všemi zastávkami a spočítáme, kolik zastávek odpovídá danému porouchanému displeji. Pokud odpovídá právě jedna zastávka, pak je jednoznačné, jaká zastávka následuje, takže vypíšeme `ANO`. V opačném případě vypíšeme `NE`.

Samotné porovnání, jestli zastávka odpovídá porouchanému displeji, je jednoduché. Stačí zkontrolovat, jestli jsou všechny rozsvícené pixely na porouchaném displeji rozsvícené i na displeji zastávky.

Časová složitost je $O(RSMN)$. Máme totiž M porouchaných displejů a každý porovnáváme s N zastávkami. Zároveň má každý displej $R{\times}S$ pixelů, které musíme v nejhorším případě projít.

Program (Python 3):
`http://ksp.mff.cuni.cz/viz/34-Z3-2.py`

Úlohu připravili: Petr Budai, Kuba Pelc, Lucka Vomelová

34-Z3-3 Tabulky a olympiáda

Využijeme každý vzoreček, na který narazíme a jehož vstupy známe a výsledek nikoliv. Až spočítáme výslednou veličinu, tak můžeme program ukončit.

Budeme opakovaně procházet seznam vzorečků. U každého vzorečku budeme zjišťovat, zda je použitelný, tedy jeho vstupní veličiny jsou známé a výstupní veličina není. U takového vzorečku si jeho výslednou veličinu označíme jako známou a daný vzoreček vypíšeme na výstup. Procházení ukončíme, pokud prohlásíme za známou veličinu X, nebo jsme při posledním průchodu nevypsali žádný vzorec. Do té doby procházíme vzorečky pořád dokola.

Již spočítané veličiny si můžeme ukládat do nějaké množiny, například hashovací tabulky nebo vyvažovaného vyhledávacího stromu.

Jelikož po nás zadání nechce nejkratší možný výpočet, je toto řešení korektní. Když vypočítáme něco navíc, tak to rozhodně nevadí a je jedno, jak ke spočítání každé veličiny dojdeme. Kdybychom se zastavili, aniž bychom znali veličinu X, zbytek vzorců, včetně vzorců s výstupem X, nejde spočítat se zadanými veličinami. Tento případ však nenastane – zadání zaručuje, že řešení vždy existuje.

Jak je naše řešení rychlé? Označme N počet na začátku známých veličin a M počet vzorečků. Na začátku musíme vložit známé veličiny do množiny. Každým průchodem seznamu vzorečků spočítáme alespoň jednu novou veličinu, průchodů tedy bude nejvýše M. V každém průchodu vyhodnocujeme M vzorečků, každý pomocí konstantně mnoha operacemi s množinou (vložení či dotazů). Celkem je tedy potřeba $\mathcal{O}(N + M^2)$ množinových operací.

Časová složitost operace s množinou je $\mathcal{O}(1)$ průměrně (pro hashovací tabulku) resp. $\mathcal{O}(\log(N + M))$ v nejhorším případě (vyhledávací strom). Celková složitost tedy je $\mathcal{O}(N + M^2)$ resp. $O((N + M^2)\log(N + M))$. Kdyby byly vzorečky na vstupu seřazené v opačném pořadí, než by bylo potřeba, tak náš algoritmus skutečně v každém průchodu použije jen jeden a tedy uvedené odhady nelze zlepšit.

Tedy budeme muset zlepšit algoritmus! Všimneme si, že v každém průchodu má smysl dívat se jen na vzorečky, u kterých jsme alespoň jednu veličinu spočítali v minulém průchodu. Tím sice vzorečky, které bychom spočítali ve stejném průchodu jako poslední z jejich vstupu, odložíme na další průchod, ale to nevadí. Jinak se nic nezmění.

Jen vynecháme vzorečky, které již mají oba dva vstupy dávno spočítané (a tedy i jejich výstup je již taky spočítaný) a vzorečky, které ještě nemají spočítané oba dva vstupy.

Na začátku algoritmu si tedy jen připravíme pro každou veličinu seznam vzorečků, které ji využívají. Jedním průchodem vzorečků tyto seznamy naplníme. Pak si vždy budeme pamatovat veličiny spočtené v minulém kole a vždy projdeme jen jejich příslušné seznamy. Na začátku projdeme seznamy známých veličin.

Každý seznam projdeme jen jednou, protože každou veličinu spočítáme jen jednou. Každý vzoreček tedy uvidíme nanejvýš dvakrát. Celková časová složitost tedy bude $\mathcal{O}(N+M)$ resp. $O((N+M)\log(N+M))$ v závislosti na implementaci množiny.

Poznámka: V odhadech složitosti byla pro jednoduchost délka názvů veličin považovaná za konstantní. Pro přesnější odhad složitosti by bylo nutně uvažovat délku veličin v množinových operacích a kontrole vzorců.

Program (Python 3):
`http://ksp.mff.cuni.cz/viz/34-Z3-3.py`

Úlohu připravili: Vojta Káně, Jirka Sejkora

34-Z3-4 Velikost plachet

Možností, jak rozvěsit plachty, je až faktoriál počtu ráhen, tedy i počtu stěžňů. Budeme tedy muset přijít s chytřejším řešením i za cenu složitějšího důkazu správnosti.

Intuitivně se nabízí párovat nejdelší ráhna s nejdelšími stěžni. Čím větší konstanta a, tím rychleji lineární funkce ax roste. Dává tedy smysl z největšího možného a vytěžit co nejvíce, a pak teprve řešit pomaleji rostoucí funkce. Myšlenku máme, pojďme ji tedy přetavit ve formální důkaz.

Nechť R je délka nejdelšího ráhna a S délka nejdelšího stěžně. Ukážeme, že mezi optimálními řešeními určitě existuje takové, kde R a S jsou v páru. Podívejme se na jedno z optimálních řešení: V něm existují páry Rs a rS pro nějaký stěžeň s a nějaké ráhno r. Ukážeme si, že nahrazením těchto párů za RS a rs se řešení nezhorší. Tedy, že platí:

$$RS + rs \geq Rs + rS$$

To můžeme dále upravovat:

$$RS - rS - Rs + rs \geq 0$$

$$S(R-r) - s(R-r) \geq 0$$

$$(R-r)(S-s) \geq 0$$

Ekvivalentními úpravami jsme dospěli k pravdivé rovnici, protože $R \geq r$ a $S \geq s$, tedy levá strana je součinem nezáporných závorek.

Teď víme, že R a S jsou určitě spolu v páru. Proto je můžeme odebrat a se zbytkem řešit úlohu identicky.

Na závěr poznamenejme, že opakované hledání nejdelších součástí se dá usnadnit tím, že si stěžně i ráhna na začátku seřadíme. Na to nám bude stačit čas $\mathcal{O}(n \cdot \log n)$ a paměť $\mathcal{O}(n)$, kde n je počet plachet. Do stejné složitosti se vejdeme i s lineárním průchodem přes seřazené součásti a jejich párováním.

Úlohu připravili: Vojta Káně, Eliška Vítková

Výsledková listina čtvrté série začátečnické kategorie 34. ročníku KSP

	řešitel	*škola*	*ročník*	*sérií*	*Z3-1*	*Z3-2*	*Z3-3*	*Z3-4*	*série*	*celkem*
0.					8	10	12	14	44,0	132,0
1.	Richard Tichý	SG Kladno	0	3	8	10	12	14	44,0	132,0
2.	Zuzana Aubrechtová	GHeyrovPH	3	3	8	10	12	10	40,0	128,0
3.	Adam Jahoda	GKepleraPH	3	5	8	10	12	13	43,0	124,0
4.	Kryštof Maxera	GJírovcČB	1	10	8	10	12	8	38,0	123,5
5.	Anna-Kristina Migel	GNAlejíPH	−1	3	8	10	12	7	37,0	123,0
6.	Jakub Hampl	GMělník	2	3	8	10	4	14	36,0	121,5
7.	Jáchym Kouba	GJŠkodyPŘ	2	3	8	10	8,7	8	34,7	120,2
8.	Jakub Mikeš	GJŠkodyPŘ	4	6	8	10	4	14	36,0	119,0
9.	Kryštof Latka	PORG Krč	4	6	8	10	4	13	35,0	118,5
10.	Ján Plachý	G VBN Prie	4	3	8	10	12		30,0	115,5
11.	Lukáš Linek	GOpatovPHA	−2	3	8	10	12		30,0	112,0
12.	Alexandr Bihun	GJírovcČB	2	3	8	10	7	8	33,0	102,5
13.	Lukáš Létal	GJŠkodyPŘ	3	8	8	10	4	13	35,0	102,0
14.	Jakub Smolík	GEBenešeKL	4	7	8	10			18,0	101,5
15.	Václav Kouřil	GTachov	4	3	8	10	7	6	31,0	99,0
16.	Vladimír Sklenár	GTerVans	2	3	8	10	4		22,0	96,0
17.	Matěj Hošek	GVolgogrOS	0	4	8	10	4		22,0	95,0
18.	Tomáš Janovec	GMnichHrad	4	9	8				8,0	94,5
19.	Viktor Číhal	SPŠSmíchov	2	5	8				8,0	93,5
20.	Vít Olšovec	GPřípotoPH	0	3	8	1	12		21,0	92,0
21.	Matúš Púll	GZborovPH	2	3	8	10	12		30,0	90,0
22.–24.	Adam Kolník	SSŠVTPraha	3	7	8	10	12		30,0	88,0
	Vojtěch Lančarič	SPŠG Třebešín	3	2	8	10	12	14	44,0	88,0
	Šimon Šustek	G Brandýs	4	2	8	10	12	14	44,0	88,0
25.	Viktor Helmich	GTMannaPH	3	2	8	10	12	13	43,0	86,0
26.	Daniel Šoltýs	GTřeKošice	4	6	8	10	4		22,0	83,0
27.	Oto Skýpala	GJŠkodyPŘ	−2	6	8	10		7	25,0	82,5
28.	Jakub Kopčil	GMikulášPL	3	6	8	10		13	31,0	82,3
29.	Jakub Podskalský	SSŠVTPraha	2	3	8				8,0	81,0
30.	Jáchym Löwenhöffer	GEvolutionJM	1	3	8	10	2,3	10	30,3	80,3
31.–32.	Tomáš Pražák	GJSeiferPH	1	5	8	10	4		22,0	80,0
	Olga Cinková	ArcibisGPH	2	8		10		10	20,0	80,0
33.–34.	Matúš Duchyňa	GGrössBA	3	3	8	10			18,0	79,0
	Ivan Trenčanský	GLSáru	3	3	8	10			18,0	79,0
35.	Nikolay Fomichev	SSŠVTPraha	3	3	8	10	12		30,0	78,0
36.	Svatava Šimečková	GJarošeBO	0	4	8		12	14	34,0	77,0
37.	Jan Prosecký	GNoMěsNMor	3	3	8			7	15,0	76,0
38.	Jan Černohorský	G Brandýs	4	2	8	10	12		30,0	73,5
39.	Bobur Toshtemirov	GMikulášPL	3	2	8	10	12	9	39,0	70,0
40.	David Pacák	G Brandýs	1	2					0,0	69,5
41.	Tadeáš Zíka	SPŠSmíchov	1	3	8	10			18,0	65,0
42.	Thomas Riedle	BRG APP	3	12	8			7	15,0	63,5
43.	Alexandra Seďová	GVídeňskBO	1	3	8	10			18,0	62,0
44.	Štěpán Fröde	GDobruška	2	2					0,0	61,0
45.	Šimon Durda	PORG Ostrava	1	2	8	10	12	10	40,0	59,0
46.	Matěj Strnad	SPŠJičín	1	8	8				8,0	57,0
47.	Jaromír Obitko	ZS6 Kladno	0	2					0,0	56,0
48.	Jan Straka	VOŠ Ždár	2	3	8	1			9,0	55,0
49.	Jan Hlavsa	GMělník	4	7					0,0	54,0
50.–51.	Jiří Kruchina	GČeskoliPH	4	3	8	10	4		22,0	53,0
	Jan Šuráň	GZborovPH	4	2	8	10	4		22,0	53,0
52.	Vít Mitáš	GPolička	0	3	8	0	2	6	16,0	52,0
53.	Michal Budai	G JGJ PH	−3	2	8				8,0	49,0
54.	Ondřej Stupka	GVolgogrOS	2	2					0,0	47,0
55.	Petr Starý	GJírovcČB	0	3	8	0			8,0	45,0

	řešitel	*škola*	*ročník*	*sérií*	*Z3-1*	*Z3-2*	*Z3-3*	*Z3-4*	*série*	*celkem*
56.	Erik Sabol	GČeskoliPH	2	10					0,0	43,0
57.–59.	Stanislav Kozák	G Holice	4	2					0,0	42,0
	Marek Maškarinec	SPŠEMasLI	1	4					0,0	42,0
	Michal Mík	SSŠVTPraha	1	3	8	10			18,0	42,0
60.–61.	Adam Kuča	PORG Krč	4	1					0,0	41,0
	Vojtěch Venzara	GMělník	4	7					0,0	41,0
62.	Veronika Jůzková	MensaG	4	12	2,7				2,7	40,7
63.–66.	Radek Bláha	GČeskáČB	0	6	8	10	4		22,0	40,0
	Filip Neubauer	AkademGPH	2	1					0,0	40,0
	Michal Pavlíček	MendelGOP	4	1					0,0	40,0
	Jáchym Tuma	G FrýdlNOs	1	3	8				8,0	40,0
67.–69.	Petr Hladík	GMikulášPL	4	7	8	10	7	13	38,0	38,0
	Honza Kocourek	ParkLane	2	2					0,0	38,0
	Kostia Kolomiiets	GVoděraPH	2	1	8	10	12	8	38,0	38,0
70.	Alexander Mateides	GJirsíkaČB	3	5					0,0	36,0
71.	Samuel Dembinný	SPŠ Kladno	1	2			4		4,0	32,0
72.–77.	Pavel Altmann	GMikulášPL	3	8					0,0	30,0
	Dominik Dembinný	ZŠMR Kladno	−2	1	8	10	12		30,0	30,0
	Kryštof Marek	SGPCE	2	4					0,0	30,0
	Jakub Nevařil	G UherBrod	4	13					0,0	30,0
	Nikol Poláková	GMetodovaBA	3	1					0,0	30,0
	Vojtěch Skyba	G UherBrod	4	4					0,0	30,0
78.	Filip Šimek	GTurnov	3	2	8	10	4	6	28,0	28,3
79.	Milan Savickij	SPŠSmíchov	2	1					0,0	28,0
80.–84.	Viktor Čubík	G UherBrod	4	2					0,0	18,0
	Šimon Hanák	CMG Brno	−1	2	8				8,0	18,0
	Miroslav Kolouch	GJírovcČB	2	1	8	10			18,0	18,0
	Petr Kroča	G UherBrod	1	8					0,0	18,0
	Lída Pavelková	GPatočkyPH	2	1					0,0	18,0
85.–86.	Matěj Kříž	GDašickáPA	4	1					0,0	16,0
	Arnošt Polák	PORG Krč	4	2					0,0	16,0
87.	Robin Kovar	GPŠ Praha	0	1					0,0	12,0
88.	Adam Bureš	SPŠ Přerov	2	2	8				8,0	10,0
89.	Štěpán Remeš	GZborovPH	2	1	8	1			9,0	9,0
90.–96.	Adam Húšťava	EupSchoolLux	4	15					0,0	8,0
	Zara Karakaya	TAPoprad	4	1					0,0	8,0
	Jan Klokan	SPŠChom	1	1	8				8,0	8,0
	Ota Macourek	GZborovPH	2	1	8				8,0	8,0
	Lenka Poljaková	GJŠkodyPŘ	2	1	8				8,0	8,0
	Jozef Remiš	G Bilíkova	3	1	8				8,0	8,0
	Marek Švajda	G UherBrod	4	1					0,0	8,0
97.	Roman Fiala	GChomutov	4	1					0,0	7,0
98.	Marek Plachý	GJatečníÚL	3	1					0,0	6,0
99.–102.	Albert Bakoč	GZborovPH	1	1					0,0	4,0
	Jáchym Hájek	GBNěmcovHK	−1	4					0,0	4,0
	Robert Klimt	GDobříš	2	1					0,0	4,0
	Matyas Oliva	G UherBrod	4	2					0,0	4,0
103.–106.	Radim Guichen	GJírovcČB	0	1					0,0	2,0
	Janek Hlavatý	GJirsíkaČB	3	22					0,0	2,0
	Jan Koška	GJírovcČB	2	8			2		2,0	2,0
	Kateřina Vomelová	GÚstavníPH	2	1					0,0	2,0
107.	Yahor Herashchanka	GTurnov	1	5		1			1,0	1,0

KSP pro vás připravují studenti Matematicko-fyzikální fakulty Univerzity Karlovy.

Webové stránky: *https://ksp.mff.cuni.cz/*

E-mail: *ksp@mff.cuni.cz*

Organizátoři a kontakty: *https://ksp.mff.cuni.cz/kontakty/*