

Korespondenční Seminář z Programování

ZAČÁTEČNICKÁ KATEGORIE

33. ročník

KSP-Z

Květen 2021

Řešení páté série začátečnické kategorie 33. ročníku KSP

33-Z5-1 Vesmírná loď

Konečnou pozici vesmírné lodi zjistíme tak, že budeme postupně počítat pozice lodi při každé změně rychlosti. Začneme na pozici $[0, 0]$ a spočítáme posuny lodi až do času T .

Na začátku máme zadanou rychlost lodi v_x, v_y . Nyní pro n změn rychlosti spočítáme:

Zjistíme čas c , po který loď touto rychlostí poletí:

$$c = t_i - t_{i-1},$$

kde t_i je čas i -tého manévru. Na začátku dosadíme $t_0 = 0$.

Následně spočítáme novou pozici lodi. Stanovíme změnu v x -ové i y -ové (z_x, z_y) souřadnici a změnu přičteme k minulé pozici (x, y) :

$$z_x = c \cdot v_x, \quad z_y = c \cdot v_y$$

$$x += z_x, \quad y += z_y$$

Po každé změně rychlosti spočítáme novou rychlost v obou směrech:

$$v_x += d_x, \quad v_y += d_y.$$

Po dokončení výpočtu nám zbývá zjistit pozici lodi v čase T . Známe její rychlost a pozici po posledním manévru. Stačí počítat čas, po který loď poletí touto rychlostí, zjistit změnu souřadnic a to přičíst k poslední pozici lodi. Potom budeme znát finální pozici lodi.

Program (Python 3):

<http://ksp.mff.cuni.cz/viz/33-Z5-1.py>

*Úlohu připravili: Jirka Sejkora,
Jirka Setnička, Tom Sláma, Lucka Vomelová*

33-Z5-2 Přerovnávání čísel

Naším úkolem je seřadit čísla na vstupu tak, aby žádná tři po sobě následující čísla netvořila rostoucí, nebo klesající posloupnost.

Jedním z možných přístupů je postupně generovat všechna možná uspořádání (permutace) čísel na vstupu, a pro každou vygenerovanou permutaci zkontrolovat, zda neobsahuje žádná tři po sobě jdoucí klesající či rostoucí čísla. Algoritmus skončíme ve chvíli, kdy nalezneme první permutaci splňující výše uvedené podmínky.

Tento přístup má však jeden háček. Existuje celkem $N!$ permutací vstupních čísel a většina těchto permutací je špatná – nesplňuje výše uvedené podmínky. Během našeho algoritmu tedy strávíme nejvíce času generováním posloupností, které následně zahodíme.

Dalším možným řešením je kontrolovat permutaci již při jejím generování. V případě porušení zadání pak můžeme permutaci zamítnout ještě předtím, než ji celou vygenerujeme. Při generování využijeme rekurzivní přístup, kdy budeme zkoušet budovat posloupnost přidáváním čísel ze vstupu

na konec posloupnosti. Pokud přidané číslo poruší zadání, vrátíme se v rekurzi o krok zpět. Uvědomte si, že to nemusí znamenat odebrání jednoho čísla, ale třeba odebrání několika čísel (v krajním případě i zahození celé dosud vybudované posloupnosti), pokud jsme vyzkoušeli, že permutace s naším aktuálním začátkem nedokážeme správně dokončit.

Obě předchozí řešení se snaží řešit problém hrubou silou, neboť systematicky prochází prostor všech možných řešení problému, který může být velmi velký. Nedokážeme najít nějaký systém v tom, jak takové „správné“ permutace vypadají? Můžeme si všimnout, že posloupnost splňující zadání je vlastně tvořena střídáním „malých“ a „velkých“ čísel. Jak si ale takové dvě skupiny čísel vytvořit? Nejjednodušší variantou, která navíc programátorovi ušetří případnou další kontrolu nad posloupností čísel, je seřadit si posloupnost vstupních čísel vzestupně a rozdělit seřazenou posloupnost na dvě stejně velké skupiny. Pokud máme lichý počet čísel, bude první skupina obsahovat o jedno číslo více. Následně budeme střídát vypisování čísel z první a z druhé skupiny.

Otázkou zůstává, zda bude takto vytvořená posloupnost splňovat zadání? Můžeme si všimnout, že pro každé číslo z druhé skupiny platí, že je větší než libovolné číslo z první skupiny. Zároveň pro každé číslo z první skupiny platí, že je menší než libovolné číslo z druhé skupiny. Pokud tedy za sebe seřadíme číslo z první skupiny, číslo z druhé skupiny a následně číslo z první skupiny, nikdy nemůžeme dostat tři po sobě jdoucí rostoucí čísla. Obdobné tvrzení platí pro trojici sestávající z čísla z druhé skupiny, první skupiny a druhé skupiny.

Nejvíce časově náročnou operací je seřazení pole, které zabere $\mathcal{O}(N \log N)$, výpis čísel totiž zvládneme v čase lineárním s velikostí vstupu.

Program (Python 3):

<http://ksp.mff.cuni.cz/viz/33-Z5-2.py>

*Úlohu připravili: Maruška Kalousková,
Jirka Setnička, Tom Sláma, Klárka Tauchmanová*

33-Z5-3 Hledání min v chodbě

Když bychom hráli hledání min, snažili bychom se umístění min odvodit logicky. Nás program může postupovat stejně. Nejprve si představme, že víme, zdali se nachází mina pod prvním políčkem. Zbytek již dokážeme odvodit.

Číslo nad prvním políčkem udává počet min pod prvním a pod druhým políčkem. Jenže my už víme, kolik min je pod prvním políčkem. Rozdíl těchto hodnot nám řekne počet min pod druhým políčkem.

Pokračujeme druhým políčkem, kde využijeme podobný postup. Číslo nad ním nám řekne celkový počet min pod třemi políčky. Pro první dvě z nich už víme, jestli pod nimi miny jsou nebo ne. Dopočítáme, jestli je mina i pod posledním políčkem.

Takto postupně vyplňujeme zleva doprava. Vždy známe informace o předchozích dvou políčkách a z toho jsme schopni dopočítat další.

Jak však vyřešíme první políčko? U něj si můžeme jednoduše tipnout. Pokud bychom si tipli špatně, v průběhu dalšího vyplňování by nám vyšlo, že pod nějakým políčkem musí být 2 miny nebo -1 min. V takovém případě začneme odzovu, jen zvolíme druhou možnost.

Číslo nad posledním políčkem nám už ve vyplňování nijak nepomůže, o existenci poslední miny se totiž rozhodneme již na předchozím políčku. Musíme však ještě dodatečně zkontrolovat, že je správně. Mohlo by se totiž stát, že není, to by také znamenalo, že jsme první minu tipli špatně.

Program (Python 3):

<http://ksp.mff.cuni.cz/viz/33-Z5-3.py>

*Úlohu připravili: David Klement,
Jirka Sejkora, Jirka Setnička*

33-Z5-4 Záznam cesty

V našem řešení se budeme inspirovat prohledáváním do šířky (více v grafové kuchařce).¹ Nejprve si ke každému vrcholu (křižovatce) poznamenejme číslo 0. Dále si pořídíme frontu. Každý její prvek bude obsahovat dva záznamy, a sice nějaký vrchol našeho grafu a pořadí v posloupnosti kódů

(pozpátku), kdy jsme se ve vrcholu mohli nacházet. Zprvu ve frontě bude jediný prvek, cílový vrchol s nulou.

Teď budeme odebírat prvky z fronty a posouvat se příslušným způsobem, pozpátku, v posloupnosti kódů. Pro každý takto odebraný vrchol se podíváme na vrcholy, do kterých z něj vedou hrany. Takové vrcholy, které jsou označené aktuálním kódem v naší posloupnosti, přidáme do fronty a přepíšeme jejich číslo na aktuální pořadí v posloupnosti. Pokud už vrchol aktuálním pořadím označený byl, neděláme nic, abychom pro daný krok měli ve frontě každý odpovídající vrchol jen jednou. Všimněme si, že s takovým postupem budou čísla pořadí ve frontě jediné stoupat. Nestane se, že bychom po odebrání vrcholu čtvrtého kroku narazili ještě na vrchol kroku třetího.

Jakmile ve frontě narazíme na vrchol s pořadím, které už odpovídá začátku cesty dolem, stačí vypsát všechny možné počáteční vrcholy. Celkově můžeme říct, že jsme provedli K kroků a v každém jsme až z N vrcholů navštívili až M hran, čímž dojdeme k časové složitosti $\mathcal{O}(K(N + M))$. Lepší odhad nenajdeme, protože tato mez se nabyde pro úplný graf (kde z každé křižovatky vede chodba do každé jiné) se stejným kódem ve všech křižovatkách.

Úlohu připravili: Martin Koreček, Jirka Setnička

¹ <http://ksp.mff.cuni.cz/viz/kucharky/grafy>

Výsledková listina páté série začátečnické kategorie 33. ročníku KSP

	<i>řešitel</i>	<i>škola</i>	<i>ročník</i>	<i>sérií</i>	Z5-1	Z5-2	Z5-3	Z5-4	<i>série</i>	<i>celkem</i>
0.					8	10	12	14	44,0	220,0
1.	Jakub Ondroušek	GTomkovaOL	1	13	8	10	12	13	43,0	212,5
2.	Robert Jaworski	GÚstavníPH	3	22	8	10	12		30,0	206,0
3.	Kryštof Maxera	GJirovcČB	0	7	8	10	12	12	42,0	198,0
4.	Dominik Farhan	GMikulášPL	4	8	8	10			18,0	188,5
5.	Jonáš Dej	G Wicht	2	5	8	10	12	3	33,0	184,0
6.	Jan Hlavsa	GMělník	3	5	8	10	8	3	29,0	179,0
7.	Tomáš Janovec	GMnichHrad	3	6	8	10	12		30,0	164,0
8.	Vojtěch Venzara	GMělník	3	5	8	10	12		30,0	154,0
9.	Thomas Riedle	BRG APP	2	9	8	10			18,0	146,0
10.	Veronika Jůzková	MensaG	3	9	8	10	12		30,0	137,0
11.	Erik Sabol	GČeskoliPH	1	8	8	10	1		19,0	135,0
12.–13.	Robert Gemrot	GKomHavíř	4	13					0,0	134,0
	Adam Kolník	SSŠVTPraha	2	4	8	10	12		30,0	134,0
14.	Lukáš Létal	GJŠkodyPŘ	2	5	8	10	12		30,0	131,0
15.	Klára Grinerová	GZborovPH	4	5	8	7	1		16,0	126,0
16.–17.	Kryštof Latka	PORG Krč	3	3	8	10	12	14	44,0	125,0
	Michal Mlynář	GJSeiferPH	0	3	8	10	12	7	37,0	125,0
18.	Olga Cinková	ArcibisGPH	1	5	8	10	12	11,5	41,5	123,0
19.	Martin Fof	MendelGOP	3	5	8	10	12		30,0	122,0
20.	Pavel Šrytr	GMělník	4	5	8				8,0	118,0
21.–22.	Oto Skýpala	GJŠkodyPŘ	–3	3	8	10	12	7	37,0	117,0
	Jakub Smolik	GEbenešeKL	3	4	8	10	12	4	34,0	117,0
23.	Jakub Mikeš	GJŠkodyPŘ	3	3	8	10	12	9,5	39,5	116,0
24.	Pavel Altmann	GMikulášPL	2	7	8	10	12		30,0	98,0
25.	Ondřej Piroutek	GČeskoliPH	3	5	3	10		10,5	23,5	93,5
26.	Yahor Herashchanka	ZŠ Turnov	0	4					0,0	90,6
27.	David Holas	SPŠEMasLI	1	2					0,0	88,0
28.	Kryštof Marek	SGPCE	1	3	8	10	12		30,0	75,0
29.	Michal Bravanský	GBílovec	3	12					0,0	67,0
30.	Adam Húšťava	EupSchoolLux	3	14	8	10	12		30,0	65,0
31.	Vojtěch Skyba	G UherBrod	3	3		10	1		11,0	60,0
32.	Vít Novotný	GMělník	3	2					0,0	57,0
33.	Jan Soukup	GJiríPoděb	3	4	8				8,0	54,0
34.	Daniel Mencl	GMělník	3	3					0,0	53,0
35.–38.	Viktor Číhal	SPŠSmíchov	1	2	8	10	12		30,0	48,0
	Jáchym Hájek	ZŠTřebechovice	–2	3	8	10			18,0	48,0
	Petr Hladík	GMikulášPL	3	6	8	10	12		30,0	48,0
	Tomáš Pražák	GJSeiferPH	0	2	8	10	12		30,0	48,0
39.	Jonáš Koziorek	GSOŠ FrMís	4	3	5				5,0	47,0
40.	Štěpán Filipčík	GTomkovaOL	3	2					0,0	43,0
41.	Jakub Nevařil	G UherBrod	3	12					0,0	40,0
42.–43.	Matouš Bohoněk	GMělník	3	2					0,0	35,0
	Michal Žáček	MensaG	4	2					0,0	35,0
44.	Zdeněk Pezlar	GJarošeBO	3	2					0,0	34,0
45.	Jonáš Bína	ZŠŠtáflovaHB	–4	4					0,0	32,7
46.–47.	Vojtech Gadurek	PORGPha	4	1					0,0	32,0
	Matěj Strnad	ZŠRiegraSM	0	5					0,0	32,0
48.–51.	Viktor Čubík	G UherBrod	3	1					0,0	30,0
	Martina Daňková	GBystrc	4	9					0,0	30,0
	Tomáš Kašpárek	G FrýdlINos	3	1					0,0	30,0
	Antonin Spaniel	GÚstavníPH	2	1					0,0	30,0
52.	Stanislav Kozák	G Holice	3	1	8	10	3	4,5	25,5	25,5
53.–54.	Jan Kotyk	G Kolín	4	1					0,0	20,0
	Antonín Musil	PORGPha	4	4	8	10			18,0	20,0
55.	Vojtěch Čermák	SPŠEMasLI	2	4					0,0	19,0

	<i>řešitel</i>	<i>škola</i>	<i>ročník</i>	<i>sérií</i>	<i>Z5-1</i>	<i>Z5-2</i>	<i>Z5-3</i>	<i>Z5-4</i>	<i>série</i>	<i>celkem</i>
56.–59.	Tomáš Chabada	SPŠEMasLI	4	5					0,0	18,0
	Adam Jahoda	GKepleraPH	2	2	0	10			10,0	18,0
	Marek Maškarinec	GFXŠaldyLI	0	3	5	10			15,0	18,0
	Matej Štencel	GPošKošice	4	4	8	10			18,0	18,0
60.–61.	Radek Bláha	GČeskáČB	–1	4					0,0	15,0
	Adam Nedoma	SPŠEMasLI	1	1	8	7			15,0	15,0
62.	Honza Kocourek	ParkLane	1	1					0,0	14,0
63.	Jan Najman	SPSEPard	4	8					0,0	10,0
64.–66.	Jakub Drobný	ŠPMNDaGB	3	1					0,0	8,0
	Jan Machourek	GBBr	0	3					0,0	8,0
	Matyas Oliva	G UherBrod	3	1					0,0	8,0
67.	Matěj Hošek	GVolgogrOS	–1	1					0,0	7,0
68.	Martin Bulíř	SPŠEMasLI	4	1					0,0	6,0