



Milý řešiteli,

vítáme Tě u 1. série úloh 5. ročníku korespondenčního semináře MoRoUS. Jako v minulých letech i nyní Ti přinášíme úlohy ze světa profesora Morouse a jeho robotických společníků.

Řešení úloh Ti umožní nejen nahlédnout do zajímavých zákoutí umělé inteligence a robotiky, ale také zúčastnit se jarního soustředění s inspirativními lidmi, obzor rozšiřujícími přednáškami a nezapomenutelnými zážitky ;) A na nejlepší řešitele čekají na konci soutěže tematické ceny!

1. série 2018/2019

Termín odeslání 1. série: 9. 12. 2018

Kam posílat řešení?

Až budeš mít řešení hotové, pošli nám je včetně všech nákresů, prográmků a prostě všeho co by nám usnadnilo opravování Tvé úlohy. Stačí, když pošleš řešení jen některých úloh nebo jejich částí, i k těm Ti pošleme komentáře a přičteme za ně body ;)

Řešení posílej nejlépe e-mailem na adresu seminar@morous.fel.cvut.cz, nebo poštou (řešení každé úlohy v tomto případě napiš na samostatný papír A4) na adresu

Korespondenční seminář Morous, Katedra kybernetiky FEL ČVUT, Karlovo náměstí 13, 121 35 Praha





Úloha č. 1: Skladiště (20 bodů)

Robotí skladiště potravin je pravoúhlá síť chodeb, ve které se nacházejí kromě obyklých kovových pilin a vazelíny i různé laskominy, jež jsou předmětem touhy některých méně zodpovědných robotů. Proto jsou na každé křižovatce instalována pohybová čidla, z jejichž záznamu lze zjistit, ve kterém časovém okamžiku byl zachycen pohyb. Bohužel, čidla byla nakoupena ve slevové akci a některá se až o pět sekund předcházejí/zpožďují. Neví se ale, která to jsou.

V sektoru B3 (označen křížkem) došlo v čase t_0 k vloupání do krabice s vanilkovými šroubky. Pachatel (robot T2D1) byl posléze dopaden v sektoru E5 (označen hvězdičkou) v čase $t_d=9$, nicméně bez šroubků.

Víme, že T2D1 se pohybuje konstantní rychlostí 1 sektor za sekundu a že se nikdy nevrátí do stejného sektoru, ve kterém už byl.

Úkol 1.1

Kudy se T2D1 mohl pohybovat, a které chodby tedy bude potřeba prohledat, zda tam laskominy neukryl?

	1				5	
Α						
В		2	\times			<i>t</i> 0.0
B C		-1	2			$t_d = 9s$
D		2 -1	4	7	8	
E				6	*	

Pozn.: Čísla v tabulce značí, ve kterých okamžicích čidlo na daných souřadnicích zachytilo pohyb. Časy jsou uvedeny v sekundách relativně k času vloupání.





Úkol 1.2

Jak bude vypadat obecný návod pro nalezení všech tras, kudy se mohl T2D1 pohybovat, máš-li časové signály z čidel?

Tvůj návod by měl úspěšně vyřešit například tento případ záznamu:

	1	2				
A						
В		\times	0	5		$t_d = 10s$
C		× 2 2	5	3		$t_d = 10s$
D		2	4	9	*	
E						

Kolik je v tomto případě možných tras, po kterých se T2D1 mohl pohybovat?

Pozn.: Předpokládej, že místo a čas loupeže a místo a čas dopadení je znám vždy, čidla mohou (jako v předchozím případě) chybovat.

Úkol 1.3

Po této závažné události byl povolán opravář. Nyní se žádná čidla nepředcházejí ani nezpožďují, některá ale nefungují vůbec. Jak mohla v tomto případě vypadat trasa, po které se pachatel pohyboval? T2D1 byl dopaden 8 sekund po loupeži.

	1		3			
A			•		-	
В			\times		4 0.	_
C		2			$t_d = 8s$	5
A B C D E			· × · 5			
E	١.		5	*		





Úkol 1.4

Jak bude vypadat obecný návod pro nalezení všech míst, kudy se mohl T2D1 pohybovat, máš-li časové signály z čidel?

Tvůj návod by měl fungovat pro tento:

		2			
A	×	1			•
В	×	2			t — 6 a
C		3			$t_d = 6s$
D		4	5		
E			*		

i tento případ:

	1	2	3	4	5	
A		· × · ·				
В		\times				+ _ 5 a
C						$t_d = 5s$
D			*			
E						

Kolik budou mít tato zadání možných řešení? Může se stát, že zadání nebude mít řešení? Pozn.: Místo a čas loupeže a místo a čas dopadení je opět znám vždy, čidla mohou (jako v předchozím případě) vynechávat.





Úkol 1.5

Nikdo nic neopraví stoprocentně. Zjistilo se, že i po zásahu opraváře se některá čidla zpožďují/předcházejí a k tomu některá zcela vynechávají. Jak v tomto případě zjistit, kudy mohl T2D1 jít?

Opět chceme obecný návod, který vyřeší všechny možné situace, např.

	1	2	3	4	5	
A	×					
В			2			4 19 a
A B C						$t_d = 12s$
D E		3				
E					*	

Jakými způsoby lze co nejdříve odhalit neplatné řešení (např. nemá smysl hledat trasy větší délky než 12)? Nad touto otázkou se opravdu zamysli. Skladiště může být mnohem větší a i když mají roboti k dispozici počítače s opravdu velkou výpočetní kapacitou, nemuseli by se výsledku dočkat.





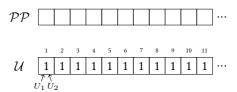
Úloha č. 2: Kouzelný počítač (20 bodů)

Kouzelný počítač se skládá ze dvou hlavních částí: Programové Pásky (PP) a Úložiště (U).

 \mathcal{PP} tvoří libovolný počet polí, každé pole obsahuje maximálně jeden příkaz. Příkazy na \mathcal{PP} se vykonávají zleva doprava.

 $\mathcal U$ tvoří také libovolný počet polí. Pole zde ale neobsahují příkazy, nýbrž celá čísla. Každé pole na začátku obsahuje číslo 1. Každé pole má svůj index, který určuje jeho pořadí (indexujeme od 1).

K \mathcal{U} také patří dva ukazatele, které odkazují na některé pole z \mathcal{U} . Na začátku oba ukazatele odkazují na první pole. Pole, na které ukazuje ukazatel 1, budeme nazývat U_1 . Pole, na které ukazuje ukazatel 2, budeme nazývat U_2 .



Počítač je připojen také k tiskárně, která umožňuje tisknout papíry s čísly.

- Nejjednodušší příkaz, který lze zapsat do \mathcal{PP} , je příkaz Print (P). Dělá pouze to, že spustí tiskárnu, která vytiskne papír s číslem v U_1 .
- Zajímavější příkaz je Add (A). Vykonání příkazu vypadá následovně: $U_1:=U_1+U_2$ (tedy do pole, na které ukazuje ukazatel 1, přičteme hodnotu z pole, na které ukazuje ukazatel 2. Všimněte si, že jde o operátor přiřazení, hodnota U_2 se tedy nezmění.)

Pokud by tedy naše \mathcal{PP} vypadala takto:

$$\mathcal{PP}$$
 P A P

bude běh programu následovný:

- 1. Print vytiskne hodnotu, na kterou ukazuje U_1 , tedy 1.
- 2. Add do pole, na které ukazuje U_1 , přičte hodnotu pole, na které ukazuje U_2 . (1+1 = 2 a v prvním poli bude tedy vepsána 2)
- 3. Print opět vytiskne hodnotu, na kterou ukazuje U_1 , tedy 2.





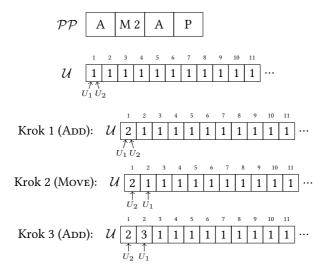
Když program skončí, budeme mít dva papíry. Na prvním bude napsané číslo 1, na druhém 2.

Dále existuje příkaz Move (M). Pomocí tohoto příkazu přesouváme ukazatele v úložišti.
Příkaz má parametr "kam", který udává, jak se ukazatele v úložišti přesunou. Může to být libovolné celé kladné číslo.

Vykonání příkazu vypadá takto:

- 1. Ukazatel 2 se přesune tak, aby ukazoval na stejné pole, jako ukazatel 1.
- 2. Ukazatel 1 se přesune tak, aby ukazoval na pole s indexem "kam".

Pokud bychom tedy chtěli vytisknout číslo 3, bude \mathcal{PP} a její provedení vypadat takto:



Krok 4 (Print): Tiskárna vytiskne papír s číslem 3

Úkol 2.1

Napište program pro kouzelný počítač, který pomocí co nejméně příkazů vytiskne číslo 100. Pro tento úkol použijte pouze výše zmíněné příkazy ADD, MOVE a PRINT.





Pokročilé příkazy

Pro počítač existuje ale mnohem více příkazů a je také připojen k numerické klávesnici. Toho hned můžeme využít.

- Protože takto skládat čísla je zdlouhavé, počítač obsahuje příkaz INPUT (IN). Ten program pozastaví a počká, než uživatel zadá na klávesnici číslo a stiskne "enter". Příkaz pak toto číslo napíše do U_1 .
- Další příkaz Subtract (S) slouží na odčítání čísel. Provedení vypadá takto: $U_1 := U_1 U_2$
- Příkaz Go (G) nám umožňuje pohybovat se po \mathcal{PP} . Příkaz má opět parametr "kam" (celé kladné číslo), který značí na kolikáté políčko \mathcal{PP} má program skočit. Program se pak začne vykonávat od daného políčka. Toto nám silně rozšiřuje obor možností. Vezměte v úvahu následující příklad:

Program se vykonává zleva doprava. Vždy, když narazí na příkaz Go 1, skočí program na první příkaz a klasicky bude dále pokračovat zleva doprava. Kouzelný počítač tak bude do nekonečna tisknout zvyšující se mocniny dvojky.

• Poslední příkaz, který budete potřebovat, je příkaz IF. Pokud U_1 je kterékoliv číslo kromě 0, příkaz neudělá nic. Program ho pouze přečte a bude pokračovat vykonáváním dalšího příkazu (příkazu, který se na \mathcal{PP} nachází vpravo od IF). Pokud U_1 je 0, program ve čtení \mathcal{PP} přeskočí první příkaz za IF a vykoná až druhý.



Program nechá uživatele zadat číslo. To se zapíše do U_1 . Pokud uživatel napsal 0, program přeskočí příkaz Go 1, přečte rovnou příkaz P, který vypíše 0. Program skončí.

Pokud uživatel zadal jiné číslo, program přečte příkaz Go 1 a program udělá skok na první pole \mathcal{PP} .





Úkol 2.2

S využitím nových příkazů, napište program, který nechá uživatele zadat číslo n (předpokládejme, že uživatel zadá kladné číslo). Pak postupně vytiskněte čísla $2,4,8,\ldots,2^n$. Program pak musí skončit.

Pozn.: V tomto případě nezáleží, kolik příkazů vaše \mathcal{PP} obsahuje.

Úkol 2.3

Nechte uživatele zadat dvě čísla (předpokládejme, že uživatel zadá dvě kladná čísla). Vytiskněte větší z nich. Pokud jsou stejná, vytiskněte obě.

Pozn.: Opět nezáleží kolik příkazů vaše PP obsahuje.

Způsob zápisu

Každý program zapisujte do samostatného textového souboru. Pořadí řádků odpovídá pořadí příkazů na \mathcal{PP} .

Příkaz	Zápis	
Add	A	
Subtract	S	
Print	P	
INPUT	IN	
Move	Mx	x je celé kladné číslo
Go	Gx	x je celé kladné číslo
IF	IF	





Témátko

Témátka můžeš odesílat v průběhu celého roku. Je jen na Tobě, jestli k němu napíšeš program, nakreslíš obrázkové řešení, vyrobíš řešení v reálu, či jen popíšeš své myšlenky. Pokus se ale vždy přijít s nějakým svým nápadem a dobře ho zdůvodni.

Témátko č. 1: Konec zdlouhavému psaní zpráv? (20 bodů)

Co kdyby počítače "uměly číst myšlenky" a napsaly zprávu za Tebe? Zamysli se nad tím, jak co nejlépe vytvořit systém, který bude uživateli nabízet slova v průběhu psaní zprávy. Jaké informace a jakým způsobem využiješ? Jak se přizpůsobit konkrétnímu uživateli?