

Meno:	Roland Vdovjak ID: 110912	Hodnotenie projektu: (max 10(TS)/5(RAM) bodov)
Cvičenie:	Streda 16:00	
Dátum:	1.5.2021	

Projekt TZIV LS2020/21 – RAM

Zadanie:	<p>18. BINÁRNE PREVODY</p> <p>Na vstupe je kladné celé číslo N (max 10000, vstup načítajte ako jedno číslo). Navrhnete programový kód pre RAM - riešenie, ktoré prevedie desiatkové číslo N do binárnej sústavy.</p>
Vstup:	<p>Akceptované vstupy: 0, 1, 2, ..., 9 999, 10 000</p> <p>Neakceptované vstupy: -100, -1, 10 001</p>
Neformálne riešenie:	<p>Pre riešenie som zvolil algoritmus, ktorý číslo zo vstupu delí konštantou 2. Ak je po delení zvyšok, tak na danom bite binárneho čísla sa nachádza číslo 1. Dolná celá časť tohto delenia je číslo, ktoré budeme pri ďalšom opakovaní algoritmu deliť.</p> <p>Každým delením je potrebné posunúť bit, na ktorý sa zapisuje 1 alebo 0. Do výsledného čísla sa čísla zapisujú postupne od najmenšieho, t.j. 2^0, 2^1 a tak ďalej. Posúva sa pomocou prenasobenia premennou 10^i, kde i predstavuje i-te opakovanie delenia konštantou 2.</p> <p>Delenie a celý proces po ňom sa opakuje pokiaľ je dolná celá časť rovná nule.</p>
Zložitosť riešenia:	<p>Algoritmus obsahuje dva cykly vnorené jeden do druhého. Prvý, hlavný riadi celý algoritmus. Druhý, vnorený násobí číslo 10 i-krát, aby sme dosiahli číslo 10^i, ktorým prenasobujeme zvyšok po delení číslom 2.</p> <p>Program po zadaní neakceptovateľného vstupu môže skončiť tromi spôsobmi, podľa vstupu. Priestorová zložitosť všetkých neakceptovateľných vstupov je</p> $S = 3$ <p>Časová zložitosť závisí od vstupu:</p> <ul style="list-style-type: none"> Vstup 0 – $T = 6$ Vstup viac ako 10 000 – $T = 8$ Záporný vstup – $T = 11$ <p>Po zadaní akceptovateľného vstupu má program priestorovú zložitosť rovnakú bez ohľadu na vstup</p> $S = 9$ <p>Program využíva osem registrov (R1 – R8), deviaty register je register R0. Úloha jednotlivých registrov je popísaná v riešení.</p>

Časová zložitosť nie je vždy rovnaká, záleží od veľkosti vstupu a taktiež od počtu bitov z výsledného čísla, ktoré majú hodnotu 1. Z toho vyplýva, že najhorší možný prípad programu je najväčšie možné číslo $2^n - 1$, keďže máme horný limit 10 000, je to číslo 8191 (bin 1 1111 1111 1111). Funkcia opisujúca *worst case* je nasledovná

$$13 + 27 * (x + 1) + \frac{x + 1}{2} * (6 + 8x)$$

Po zjednodušení

$$4x^2 + 34x + 43$$

Premenná x predstavuje dolnú celú časť $\log_2 n$, kde n je vstupné číslo.

Zložitosť sa rovná 1027.

Najlepšie prípady nastávajú v číslach 2^n , napr. číslo 8192 má časovú zložitosť 381. Najlepšie prípady sa dajú popísať vzťahom

$$26x + 43$$

Premenná x predstavuje dolnú celú časť $\log_2 n$, kde n je vstupné číslo.

Rozdiel medzi najhorším alebo najlepším prípadom je v tom, že ak po delení číslom 2 nie je zvyšok, nemusí sa rátať pozícia bitu a ani sa nepripisuje do výsledku.

Priemerná časová zložitosť je teda z intervalu medzi najlepšou a najhoršou možnou.

Ku daným funkciám reprezentujúcich najhorší a najlepší možný prípad zložitosti sme dospeli nasledovne:

INŠTRUKCIA	ZLOŽITOSŤ	
	<i>Worst case</i>	<i>Best case</i>
Read 1	1	1
Load 1	1	1
Store 7	1	1
Jump start	1	1
koniec: reject	1	1
Start: jzero koniec	1	1
Sub =10000	1	1
Jgzero koniec	1	1
Mult =-1	1	1
Jgzero koniec	1	1
Del: load 7	X+2	X+2
Jzero premenene	X+2	X+2
Load 3	X+1	X+1
Add =1	X+1	X+1
Store 3	X+1	X+1
Load7	X+1	X+1
Div =2	X+1	X+1
Store 5	X+1	X+1
Mult =2	X+1	X+1
Sub 7	X+1	X+1
Jzero par	X+1	X

		Jump nepar	X+1	1
		Par: Load =0	0	X
		Store 6	0	X
		Jump bin	0	X
		Nepar: Load =1	X+1	1
		Store 6	X+1	1
		Jump bin:	X+1	1
		Bin: Load 5	X+1	X+1
		Store 7	X+1	X+1
		Load 6	X+1	X+1
		Jzero del	X+1	X+1
		Load 3	X+1	1
		Store 8	X+1	1
		Load =1	X+1	1
		Store 4	X+1	1
		Opakuj: Load 8	(X+1)/2*(X+2)	X*1+1
		Sub =1	(X+1)/2*(X+2)	X*1+1
		Jzero zapis	(X+1)/2*(X+2)	X*1+1
		Store 8	(X+1)/2*X	X*1
		Load 4	(X+1)/2*X	X*1
		Mult =10	(X+1)/2*X	X*1
		Store 4	(X+1)/2*X	X*1
		Jump opakuj	(X+1)/2*X	X*1
		Zapis: load 2	X+1	1
		Add 4	X+1	1
		Store 2	X+1	1
		Jump del	X+1	1
		Premenene: write 2	1	1
	<p>X predstavuje dolnú celú časť $\log_2 n$, kde n je vstupné číslo. Zeleno podfarbené políčka sú pre zvýraznenie vnoreného cyklu, výpočet najhoršieho prípadu jednotkovej zložitosti vyjadruje súčet aritmetickej postupnosti. V nezjednodušenej funkcii je vidieť vzorec pre výpočet tohto cyklu, taktiež je podfarbený na zeleno</p>			
Simulátor:	SimStudio.exe, RandomAccessMachineSimulator.exe			
<p>Definícia výpočtového modelu (prechodová funkcia), kód simulátora (copy-paste):</p> <p>Kód je taktiež priložený ku dokumentácii v samostatnom súbore s názvom “Vdovjak_Ram_18.ram”.</p> <pre>//R1 Dec //R2 Bin //R3 i //R4 10^i //R5 celociselne delenie //R6 zvysook po celociselnom deleni //R7 pomocna premenna //R8 pomocna premenna</pre>				

```

read 1
load 1
store 7
jump start

koniec:
reject

// Test vstupu, pod 10000 a kladne cislo
start:
jzero koniec
sub =10000
jgzero koniec
add =10000
mult =-1
jgzero koniec

// Premena na Bin, zaciatok algoritmu ako takeho
del:
load 7
jzero premenene
load 3
add =1
store 3

load 7
div =2
store 5
mult =2
sub 7

//par == bolo parne cislo, t.j. bez zvyšku, nepar == bolo neapadne
jzero par
jump nepar

par:
load =0
store 6
jump bin

nepar:
load =1
store 6
jump bin

bin:
load 5
store 7

// ak je zvyšok nula, nie je potrebne pokracovat v iriteracii algoritmu
load 6
jzero del

// 10 na i (R3)
load 3
store 8
load =1
store 4

opakuj:
load 8

```

```
sub =1
jzero zapis
store 8
load 4
mult =10
store 4
jump opakuj
```

```
zapis:
load 2
add 4
store 2
jump del
```

```
premenene:
write 2
```