

Національний технічний університет України «Київський політехнічний інститут»

Науково-навчальний комплекс «Інститут прикладного системного аналізу»

Кафедра математичних методів системного аналізу

## ***Розрахункова робота №1***

З курсу «Математична логіка і теорія алгоритмів»

На тему «Машина Тьюрінга»

**Виконав:**

Барзій Ілля, КА-41

**Прийняв:**

Спекторський І.Я.

Київ 2015

## ***Зміст***

Постановка задачі.....	3
Обґрунтування алгоритму.....	3
Схема алгоритму.....	3
Список команд.....	4
Графічне зображення.....	7
Табличне зображення.....	8
Результати роботи програми.....	9

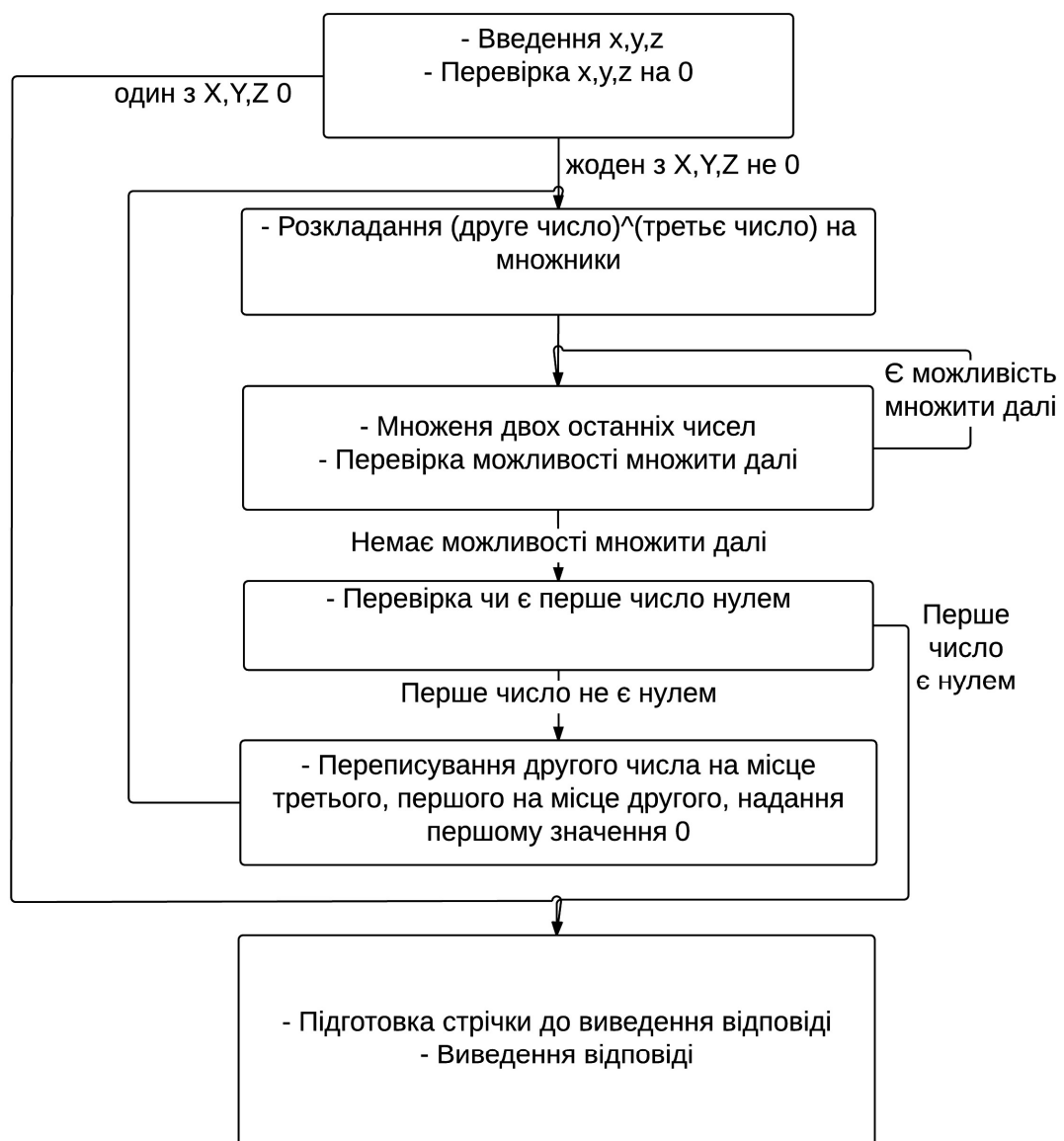
## Постановка задачі

На вхід дані числа  $x$   $y$   $z$  двійковій системі. Необхідно обчислити  $x^{y^z}$ .

## Обґрунтування алгоритму

Алгоритм полягає в тому, щоб розкласти число в степені на добуток чисел та порахувати його по черговим множенням. На початку роботи алгоритм перевіряє введену комбінацію чисел на тривіальні випадки (наявні числа 0) і, якщо такі є, одразу ж виводить відповідь. Інакше, розкладає спочатку  $y^z$  як добуток та рахує попарно. Потім зводить стрічку до вигляду, коли можна застосувати цю ж саму дію для  $x^{(y^z)}$  та виконує її. Після чого виводить результат.

## Схема алгоритму



## Список команд

### Машина перевірки нулів та одиниць

<i>start 0 _ r s0</i>	<i>s20 * _ r s20</i>	<i>s221 _ _ r p1</i>
<i>start 1 1 r s1</i>	<i>s200 _ _ l s200</i>	<i>s221 * _ l s221</i>
<i>s0 _ 1 * !</i>	<i>s200 * _ l s201</i>	<i>s220 ^ _ l s2200</i>
<i>s0 * _ r s0</i>	<i>s201 _ 1 * !</i>	<i>s2200 ^ _ l s2201</i>
<i>s1 ^ ^ r s2</i>	<i>s201 * _ l s201</i>	<i>s2200 * _ l s2200</i>
<i>s1 * _ r s1</i>	<i>s21 ^ ^ r s22</i>	<i>s2201 _ _ r !</i>
<i>s2 0 0 r s20</i>	<i>s21 * _ r s21</i>	<i>s2201 * _ l s2201</i>
<i>s2 1 1 r s21</i>	<i>s22 1 1 l s221</i>	
<i>s20 _ _ l s200</i>	<i>s22 0 _ l s220</i>	

### Машина розкладання степеня у множники

<i>p1 ^ ^ r 01</i>	<i>0c 0 8 l 0c0</i>	<i>g1 * _ l g1</i>
<i>p1 * _ r p1</i>	<i>0c 1 7 l 0c1</i>	<i>g2 _ ^ l g3</i>
<i>01 1 1 r 001</i>	<i>0c0 ^ ^ r 0c00</i>	<i>g2 * _ l g2</i>
<i>01 0 _ l 000</i>	<i>0c0 1 1 r 0c00</i>	<i>g3 _ 0 * end4</i>
<i>000 _ _ r 0000</i>	<i>0c0 0 0 r 0c00</i>	<i>; Unchanged road</i>
<i>000 * _ l 000</i>	<i>0c0 * _ l 0c0</i>	<i>o2 * _ l 1p2</i>
<i>0000 ^ _ r !</i>	<i>0c1 ^ ^ r 0c11</i>	<i>1p2 _ = r 1p3</i>
<i>0000 * _ r 0000</i>	<i>0c1 1 1 r 0c11</i>	<i>1p3 ^ ^ r 1p4</i>
<i>001 ^ ^ l 02</i>	<i>0c1 0 0 r 0c11</i>	<i>1p3 * _ r 1p3</i>
<i>001 * _ l o1</i>	<i>0c1 * _ l 0c1</i>	<i>1p4 ^ ^ r 1p5</i>
<i>o1 _ _ r o2</i>	<i>0c00 _ 0 r 0e</i>	<i>1p4 * _ r 1p4</i>
<i>o1 * _ l o1</i>	<i>0c11 _ 1 r 0e</i>	<i>1p5 _ _ l 1pa</i>
<i>; if 02 you must rewrite</i>	<i>0e 7 _ r 0c</i>	<i>1p5 * _ r 1p5</i>
<i>02 1 _ r 03</i>	<i>0e 8 _ r 0c</i>	<i>; Here Ive reached the last</i>
<i>03 ^ _ r 0c</i>	<i>0e * _ r 0e</i>	<i>symbol and begin to</i>
<i>0c _ _ l g1</i>	<i>g1 ^ ^ l g2</i>	<i>rewrite</i>
		<i>1pa 0 0 l 1pa</i>

<i>1pa 1 0 r 1pa1</i>	<i>1pad1 * *   1pad1</i>	<i>1pf * *   1pf</i>
<i>1pa ^ ^   1pf</i>	<i>1padr 8 0   1pad*?</i>	<i>1pf1 # = r 1pf2</i>
<i>1pa1 0 1 r 1pa1</i>	<i>1padr 7 1   1pad*?</i>	<i>1pf2 # # r 1pf3</i>
<i>1pa1 _ _   1pa2</i>	<i>1padr * * r 1padr</i>	<i>1pf2 * * r 1pf2</i>
<i>1pa2 ^ ^   1pad</i>	<i>1pad*? ^ ^   1pad*</i>	<i>1pf3 # #   end1</i>
<i>1pa2 * *   1pa2</i>	<i>1pad*? * * * 1pad</i>	<i>1pf3 = =   end1</i>
<i>;8 is an alternative 0 7 is an alternative 1</i>	<i>1pad* _ # r 1padb</i>	<i>1pf3 * * r 1pf3</i>
<i>1pad 0 8 r 1pad0</i>	<i>1pf2 = =   1pf0</i>	<i>1pf0 = _ r end32</i>
<i>1pad 1 7 r 1pad1</i>	<i>1pad* * *   1pad*</i>	<i>1pf0 * *   1pf0</i>
<i>1pad0 _ 0 r 1padr</i>	<i>1padb _ _   1pa</i>	
<i>1pad0 * *   1pad0</i>	<i>1padb * * r 1padb</i>	
<i>1pad1 _ 1 r 1padr</i>	<i>1pf _ _ r 1pf1</i>	

### **Машина множения**

<i>end1 1 7   q1</i>	<i>2 1 7   3</i>	<i>14 0 0 r 14</i>
<i>end1 0 8   15</i>	<i>2 0 8   6</i>	<i>14 # # r 14</i>
<i>end1 # #   17</i>	<i>; Describing == line in 2</i>	<i>14 * * * 1</i>
<i>;Describing line of 0</i>	<i>10 9 0   10</i>	<i>1 7 1   end1</i>
<i>15 _ _ * 16</i>	<i>10 6 1   10</i>	<i>1 8 0   end1</i>
<i>15 7 7   16</i>	<i>10 7 7   10</i>	<i>;Describing 1 line in 2</i>
<i>15 8 8   16</i>	<i>10 8 8   10</i>	<i>3 = =   4</i>
<i>15 * *   15</i>	<i>10 * * r 11</i>	<i>3 * *   3</i>
<i>16 1 7 r 13</i>	<i>11 1 1 r 11</i>	<i>4 1 9   5</i>
<i>16 0 8 r 13</i>	<i>11 0 0 r 11</i>	<i>4 _ 6 r 8</i>
<i>16 _ 8 r 13</i>	<i>11 * *   12</i>	<i>4 0 6 r 8</i>
<i>16 * *   16</i>	<i>12 0 8 r 13</i>	<i>4 * *   4</i>
<i>;Describing line of 1</i>	<i>12 1 7 r 13</i>	<i>5 1 0   5</i>
<i>q1 # #   2</i>	<i>13 = = r 14</i>	<i>5 0 1 r 8</i>
<i>q1 * *   q1</i>	<i>13 * * r 13</i>	<i>5 _ 1 r 8</i>
<i>2 = =   10</i>	<i>14 1 1 r 14</i>	<i>8 = = r 9</i>

8 * * r 8	7 _ 9 r 8	17 8 0   17
9 7 1   2	7 0 9 r 8	17 * *   17
9 8 0   2	7 1 6 r 8	
9 * * r 9	7 * *   7	
; Describing 0 line in 2	; Describing exit from 17	
6 = =   7	17 _ _ r end2	
6 * *   6	17 7 1   17	

### **Машина перенесення множника та вибору подальших дій**

end2 * *   3q1	3q51 * *   3q53	3qe3 _ = r 3qe4
3q1 _ = r 3q2	3q53 _ 1   3qe	3qe4 # # r 3qen1
3q2 # # r 3q3	3q50 _ _ r 3q50	3qe4 = =   3qen2
3q2 * * r 3q2	3q50 * *   3q52	3qe4 * * r 3qe4
3q3 = =   3q4	3q52 _ 0   3qe	3qen1 # #   end31
3q3 # #   3q4	3qe 1 1 * 3q5	3qen1 = =   end31
3q3 * * r 3q3	3qe 0 0 * 3q5	3qen1 * * r 3qen1
3q4 = _   3q5	3qe _ _   3qe	3qen2 = _ r end32
3q4 * _   3q4	3qe = _ r 3qe2	3qen2 * *   3qen2
3q5 1 _ r 3q51	3qe2 1 1   3qe3	
3q5 0 _ r 3q50	3qe2 0 0   3qe3	
3q51 _ _ r 3q51	3qe2 * * r 3qe2	

### **Машина вибору подальших дій**

end32 = = r tq=	tq= 1 1   tq1	tq0 _ _ r endp2
end32 * * r end32	tq1 _ _ r endp1	tq0 * *   tq0
tq= 0 0   tq0	tq1 * *   tq1	

### **Машина підготовки стрічки до повторного піднесення до степеня**

endp1 * *   4q1	4q2 * * r 4q2	4q4 = +   4q5
4q1 _ = r 4q2	4q3 _ _   4q4	4q4 * *   4q4
4q2 ^ ^ r 4q3	4q3 * _ r 4q3	4q5 = = r 4qp

4q5 \* \* / 4q5

4qp1 \_ 1 / 4q5

4qd1 = ^ / 4qd2

4qp + = / 4qd

4qp1 \* \* r 4qp1

4qd1 \_ \_ r 4qd1

4qp 8 8 r 4qp

4qp0 \_ 0 / 4q5

4qd2 \_ 0 \* end4

4qp 7 7 r 4qp

4qp0 \* \* r 4qp0

4qp 0 8 r 4qp0

4qd \_ \_ r 4qd1

4qp 1 7 r 4qp1

4qd \* \_ / 4qd

### Машина підготовки стрічки до виводу відповіді

endp2 \* \* / eo1

eo2 \* \* r eo2

eq2 = \_ r !

eo1 \_ = r eo2

eq1 \_ \_ / eq2

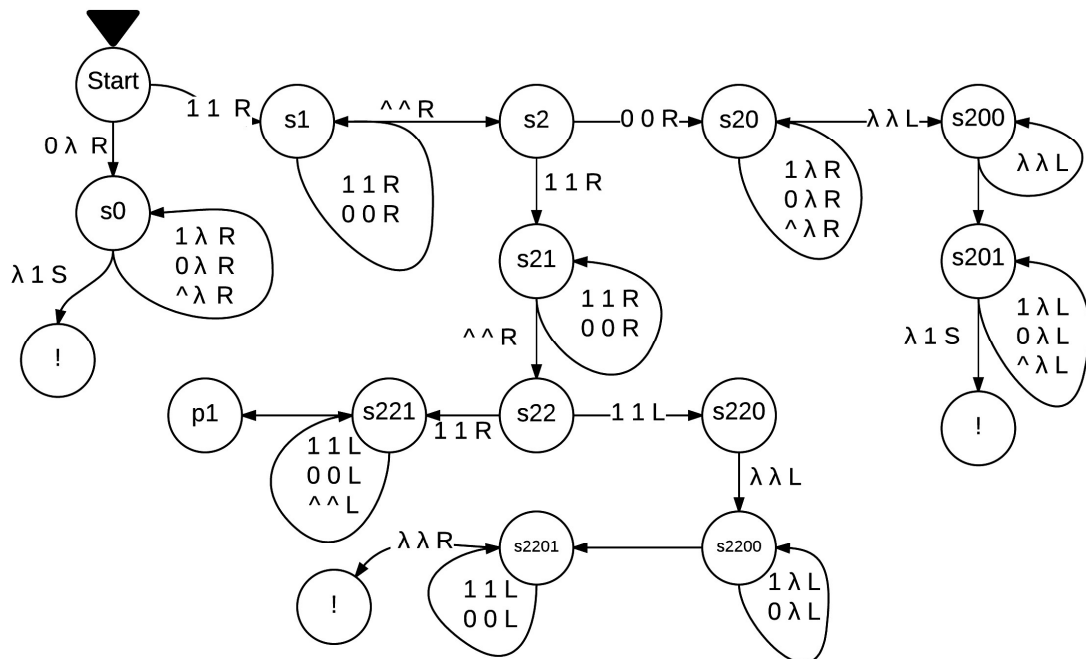
eq2 \* \* / eq2

eo2 = \_ r eq1

eq1 \* \_ r eq1

### Графічне зображення

Зображення машини Тьюрінга у вигляді орієнтованого міченого графу зобразимо на прикладі перевірки x,y,z на наявність 0.



## Табличне зображення

Зображення машини Тьюрінга у вигляді таблиці покажемо на прикладі перевірки  $x, y, z$  на наявність 0. <sup>2</sup>

	1	0	$\wedge$	$\lambda$
start	$\lambda$ s0 R	1 s1 R		
s0	$\lambda$ s0 R	$\lambda$ s0 R	$\lambda$ s0 R	1 ! S
s1	1 s1 R	0 s1 R	$\wedge$ s2 R	
s2	1 s21 R	0 s20 R		
s20	$\lambda$ s20 R	$\lambda$ s20 R	$\lambda$ s20 R	$\lambda$ s200 L
s200	1 s201 $\lambda$	0 s201 $\lambda$	$\wedge$ s201 $\lambda$	$\lambda$ s200 L
s201	$\lambda$ s201 L	$\lambda$ s201 L	$\lambda$ s201 L	1 ! S
s21	1 s21 R	0 s21 R	$\wedge$ s22 R	
s22	1 s221 L	0 s220 L		
s221	1 s221 L	0 s221 L	$\wedge$ s221 L	$\lambda$ p1 R
s220				$\lambda$ s2200 L
s2200	$\lambda$ s2200 L	$\lambda$ s2200 L	$\lambda$ s2201 L	
s2201	1 s2201 L	0 s2201 L		$\lambda$ ! R
p1				
!				



## **Результати роботи програми**

8

**Вхідні дані:  $111^{11^0}$**

**Вихідні дані: 111**

**Кількість команд: 17**

**Вхідні дані:  $101^{0^{11}}$**

**Вихідні дані: 1**

**Кількість команд: 11**

**Вхідні дані:  $10^{10^{10}}$**

**Вихідні дані: 10000**

**Кількість команд: 1436**

**Вхідні дані:  $101^{11^{11}}$**

**Вихідні дані: 1001110110010111000111100100111111010000100000000011110011101**

**Кількість команд: 260529**