

PROGRAMOZÁS

Nagyszám!

Gregorics Tibor

<http://people.inf.elte.hu/gt/prog>

Feladat

Számoljuk ki egy természetes szám faktoriálisát!

Összegzés:

m..n	~	2 .. n
f(e)	~	i
s	~	f
H, +, 0	~	\mathbb{N} , *, 1

$f := 1$

$i = 2 .. n$

$f := f * i$

$A = n : \mathbb{N}, f : \mathbb{N}$

$Ef = n = n'$

$Uf = n = n' \wedge f = \prod_{i=2}^n i$

Specifikáció alapján (Fekete doboz)

Tesztelés

❑ Érvényes tesztesetek

○ Összegzés tétele alapján:

- intervallum hossza: nulla $n = 1 \rightarrow f = 1$
egy $n = 2 \rightarrow f = 2$
több $n = 5 \rightarrow f = 120$
- intervallum eleje és vége: $n = 3 \rightarrow f = 6$
- terhelés (skálázás):

int f : 12! -ig jó

long double f : 25! -ig jó

Egy igazán nagy szám kellene!

- Határ adat, különleges adat: $n=0 \rightarrow f = 1$
 $n=1 \rightarrow f = 1$

❑ Érvénytelen tesztesetek ($n < 0$) : $n = -1, -45$

Program (kód) alapján (Fehér doboz)

❑ Beolvasás tesztelése: $n = -3.14$, hét

❑ ...

BigNumber típusa

Típus-specifikáció

típus-értékek

\mathbb{N}

$f : \text{BigNumber}, k : \mathbb{N}_{32}$

Szorzás

$f := f * k$

Értékadás

$f := k$

Kiírás

$\text{cout} := \text{cout} \oplus f \quad (\text{cout} \ll f)$

műveletek

reprezentációs függvény

Egy természetes számot a decimális számjegyeinek fordított sorrendjének sztringjével helyettesítünk:

Ötszázhetvenegy: "175"

Nulla: ""

reprezentáció

String

típus invariáns

Számjegyekből álló sorozatok $\{'0' \dots '9'\}^*$, amelyek végén nincsenek nullák.

Olyan programok, amelyek **BigNumber** típusú változók helyett **String** típusú változókkal dolgoznak, de úgy, mintha mégis nagy számokkal dolgoznának.

implementáció

Típus-implementáció

BigNumber műveleteinek implementációiról

f nagy szám kiírása cout-ra: $\text{cout} := \text{cout} \oplus f$

"175" \rightarrow 571 "" \rightarrow 0

f nagy számnak értékül adni egy nem-negatív int-et: $f := k$

oszlopérték: $\frac{571}{1} \quad \frac{57}{7} \quad \frac{5}{5} \quad \frac{0}{-}$
 maradék: \rightarrow "175"

f nagy szám szorzása egy nem-negatív int-tel : $f := f * k$

"175" \times 32

1	7	5			
<hr style="border: 0.5px solid black;"/>					
32	224	160			
<hr style="border: 0.5px solid black;"/>					
		3	22	18	1
<hr style="border: 0.5px solid black;"/>					
32	227	182	18	1	0
<hr style="border: 0.5px solid black;"/>					
2	7	2	8	1	

átvitel: \rightarrow "27281"

oszlopérték: $\frac{32}{2} \quad \frac{227}{7} \quad \frac{182}{2} \quad \frac{18}{8} \quad \frac{1}{1} \quad \frac{0}{-}$
 maradék: \rightarrow "27281"

Mindhárom esetben egy sorozatot kell létrehozni (ÖSSZEGZÉS), amelyhez egy alkalmas felsorolást kell használni.

Kiírás

$A = f : \text{BigNumber}, \text{ cout} : \text{String}$

$A = \text{str} : \text{String}, \text{ cout} : \text{String}$

$Ef = \text{str} = \text{str}' \wedge \text{str}$ kielégíti az invariánst

$Uf = \text{str} = \text{str}' \wedge (|\text{str}| > 0 \rightarrow \text{cout} = \bigoplus_{i=|\text{str}|}^1 \langle \text{str}_i \rangle) \wedge$
 $(|\text{str}| = 0 \rightarrow \text{cout} = \langle 0 \rangle))$

Összegzés:

$e \in t$	\sim	$i = \text{str} .. 1$	$i = \text{str} .. 1$		$\text{cout} := \langle 0 \rangle$
$f(e)$	\sim	$\langle \text{str}_i \rangle$	$\text{cout} := \text{cout} \oplus \langle \text{str}_i \rangle$		
s	\sim	cout			
$H, +, 0$	\sim	$\text{String}, \oplus, \langle \rangle$			

Értékadás

$A = k : \mathbb{N}_{32}, f : \text{BigNumber}$

$A = k : \mathbb{N}_{32}, \text{str} : \text{String}$

$Ef = k = k'$

$Uf = k = k' \wedge \text{str} = \bigoplus_{c \in t} \langle \text{char}(c \bmod 10) \rangle$

str kielégíti az invariánst

a 0 és 9 közötti számjegy karaktere

$c \in t$

$t : \text{enor}(\mathbb{N}_{32})$

$t = \langle k, k/10, k/100, \dots \rangle$

First: $c := k$; Next: $c := c/10$; Current: c ; End: $c=0$

Összegzés:

$e \in t$	\sim	$c \in t$
$f(e)$	\sim	$\langle \text{char}(c \bmod 10) \rangle$
s	\sim	str
$H, +, 0$	\sim	$\text{String}, \bigoplus, \langle \rangle$

$\text{str} := \langle \rangle$

$c := k$

$c \neq 0$

$\text{str} := \text{str} \bigoplus \langle \text{char}(c \bmod 10) \rangle$

$c := c/10$

Szorzás

$A = f : \text{BigNumber}, k : \mathbb{N}_{32}$

$t_1 : \text{enor}(\mathbb{N}_{32})$

$t_1 = \langle \dots \text{str}_i * k + \text{átvitel} \dots \rangle$ (oszlopértékek)

First: $i, c := 1, 0$; Next: $i := i + 1$; End: $i > |\text{str}|$;

Current: $c := \text{str}_i * k + c / 10 : c$

$A = \text{str} : \text{String}, k : \mathbb{N}_{32}$

$Ef = \text{str} = \text{str}' \wedge k = k' \wedge \text{str}$ kielégíti az invariánst

$Uf = k = k' \wedge \text{str} = \left(\bigoplus_{c \in t_1} \langle \text{char}(c) \bmod 10 \rangle \right)$

$\bigoplus \left(\bigoplus_{c \in t_2} \langle \text{char}(c \bmod 10) \rangle \right)$

$t_2 : \text{enor}(\mathbb{N}_{32})$

($|\text{str}|$ -dik oszlopérték = $t_1.\text{Current}()$, amit jelöljön c')

$t_2 = \langle c' / 10, c' / 100, \dots \rangle$

First: $c := c' / 10$; Next: $c := c / 10$; Current: c ; End: $c = 0$

str kielégíti az invariánst?

Szorzás 1.

$A = \text{str} : \text{String}, k : \mathbb{N}_{32}, c : \mathbb{N}_{32}$

$t_1 : \text{enor}(\mathbb{N}_{32})$

First: $i, c := 1, 0$; Next: $i := i + 1$; End: $i > |\text{str}|$;

Current: $c := \text{str}_i * k + c / 10 : c$

$Ef = \text{str} = \text{str}' \wedge k = k' \wedge \text{str kielégíti az invariánst}$

$Uf = k = k' \wedge \text{str} = (\bigoplus_{c \in t_1} \langle \text{char}(c) \bmod 10 \rangle) \wedge c = t_1.\text{Current}()$

$i, c := 1, 0$

$i \leq |\text{str}|$

$c := \text{érték}(\text{str}_i) * k + c / 10$

$\text{str} := \text{str} \oplus \langle \text{char}(c \bmod 10) \rangle$

$i := i + 1$

Összegzés:

$e \in t$	\sim	$c \in t_1$
$f(e)$	\sim	$\langle \text{char}(c \bmod 10) \rangle$
s	\sim	str
$H, +, 0$	\sim	$\text{String}, \oplus, \langle \rangle$

Szorzás 1. még egyszer

$A = \text{str} : \text{String}, k : \mathbb{N}_{32}, c : \mathbb{N}_{32}$

$t_1 : \text{enor}(\mathbb{N}_{32} \times \mathbb{N}_{32})$

First: $i, c := 1, 0$; Next: $i := i + 1$; End: $i > |\text{str}|$;

Current: $c := \text{str}_i * k + c / 10 : (c, i)$

$Ef = \text{str} = \text{str}' \wedge k = k' \wedge \text{str kielégíti az invariánst}$

$Uf = k = k' \wedge (\forall (c, i) \in t_1 : \text{str}_i = \langle \text{char}(c) \bmod 10 \rangle) \wedge c = t_1.\text{Current}()$

$i, c := 1, 0$

$i \leq |\text{str}|$

$c := \text{érték}(\text{str}_i) * k + c / 10$

$\text{str}_i := \langle \text{char}(c \bmod 10) \rangle$

$i := i + 1$

Összegzés:

$e \in t$	\sim	$c, i \in t_1$
$f(e)$	\sim	$\langle \text{char}(c \bmod 10) \rangle$
s	\sim	str
$H, +, 0$	\sim	$\text{String}, \oplus, < >$

Szorzás 2.

$A = \text{str} : \text{String}, \quad c : \mathbb{N}_{32}$ $t_1.\text{Current}()$

$Ef = \text{str} = \text{str}'' \wedge c = c' \wedge \text{str kielégíti az invariánst}$

$Uf = \text{str} = \text{str}'' \oplus (\oplus \langle \text{char}(c \bmod 10) \rangle)$

$c \in t_2$

$t_2 = \langle c'/10, c'/100, \dots \rangle$

First: $c := c'/10$, Next: $c := c/10$, Current : c , End: $c=0$

$c := c/10$

$c \neq 0$

$\text{str} := \text{str} \oplus \langle \text{char}(c \bmod 10) \rangle$

$c := c/10$

Összegzés:

$e \in t$

\sim

$c \in t_2$

$f(e)$

\sim

$\langle \text{char}(c \bmod 10) \rangle$

s

\sim

str

$H, +, 0$

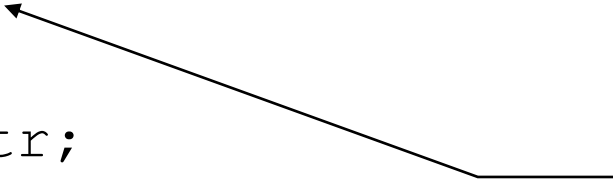
\sim

$\text{String}, \oplus, \langle \rangle$

most nem kell: $\text{str} := \langle \rangle$

BigNumber kódja

```
class BigNumber{  
private:  
    std::string str;  
  
public:  
    void Ertekadas(const int k);  
    BigNumber Szorzas(const int k);  
    void Kiiras() const;  
};
```



```
BigNumber f;  
...  
f.Ertekadas(1);  
...  
f = f.Szorzas(i);  
...  
f.Kiiras();
```

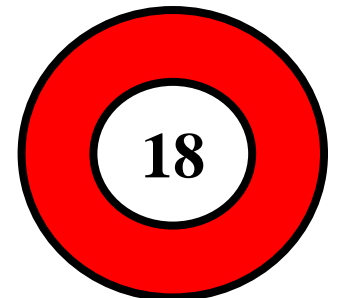
```
void BigNumber::Kiiras() const  
{  
    if(str.size()>0){  
        for(int i=str.size()-1; i>=0; --i) {  
            cout << str[i];  
        }  
    } else cout << 0;  
}
```

```
class BigNumber{  
private:  
    std::string str;  
public:  
    int operator=(const int k);  
    BigNumber operator*(const int k);  
    friend std::ostream& operator<<(  
        std::ostream &cout, const BigNumber &bn);  
};
```

BigNumber kódja másképp

```
BigNumber f;  
f = 1;  
f = f * i;  
cout << f;
```

```
ostream& operator<<(ostream &cout, const BigNumber &bn)  
{  
    if (bn.str.size() > 0) {  
        for (int i = bn.str.size() - 1; i >= 0; --i) {  
            cout << bn.str[i];  
        }  
    } else cout << 0;  
    return cout;  
}
```



Tesztelés

❑ Metódusonkénti tesztelés

- Minden metódusra elvégezzük annak fekete és fehérdoboz tesztelését.

❑ Osztály tesztelése

- BigInteger típusú objektumokat hozunk létre és azokra a metódusaik tetszőleges variációit próbáljuk ki.

unit teszt

kell egy tesztkörnyezetet : egy speciális főprogram, ami többnyire egy menü

❑ Főprogram tesztje

- lásd korábban

Hiba, ha

- negatív számot adunk értékül
- negatív számmal szorzunk
- nullával szorzunk

Szorzás újra

$$A = \text{str} : \text{String}, \quad k: \mathbb{N}_{32}$$

$$Ef = \text{str} = \text{str}' \wedge k = k' \wedge \text{str} \text{ kielégíti az invariánst}$$

$$Uf = k = k' \wedge (k=0 \rightarrow \text{str} = \langle \rangle)$$

$$\wedge (k>0 \rightarrow \text{str} = (\bigoplus_{c \in t_1} \langle \text{char}(c) \bmod 10 \rangle) \oplus (\bigoplus_{c \in t_2} \langle \text{char}(c) \bmod 10 \rangle))$$

A nullával való szorzás hibája:
str nem elégíti ki az invariánst, ha k=0