# BSA法（Binary Splitting Algorithm）

**定理1**

2 つの関数,

$$f(x)=\frac{ax+b}{cx+d}\ ,\quad g(x)=\frac{sx+t}{ux+v}$$

に対して

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} s & t \\ u & v \end{pmatrix}=\begin{pmatrix} p & q \\ r & s \end{pmatrix}$$

とすると

$$f(g(x))=\frac{px+q}{rx+s}$$

**証明**

$$f(g(x))=\frac{a\cdot g(x)+b}{c\cdot g(x)+d}=\frac{a\cdot\dfrac{sx+t}{ux+v}+b}{c\cdot\dfrac{sx+t}{ux+v}+d}=\frac{a\cdot(sx+t)+b\cdot(ux+v)}{c\cdot(sx+t)+d\cdot(ux+v)}$$

$$=\frac{(as+bu)x+(at+bv)}{(csx+du)x+(ct+dv)}=\frac{px+q}{rx+s}$$

**証明終**

**定理 2** $S_n=\frac{1}{C_0}(A_0+\frac{B_0}{C_1}(A_1+\frac{B_1}{C_2}(A_2+\frac{B_2}{C_3}(A_3+\cdots(A_{n-1}+\frac{B_{n-1}}{C_n}(A_n))))))$

$$\begin{pmatrix} B_0 & A_0 \\ 0 & C_0 \end{pmatrix}\begin{pmatrix} B_1 & A_1 \\ 0 & C_1 \end{pmatrix}\begin{pmatrix} B_2 & A_2 \\ 0 & C_2 \end{pmatrix}\cdots\begin{pmatrix} B_n & A_n \\ 0 & C_n \end{pmatrix}=\begin{pmatrix} T & P \\ 0 & Q \end{pmatrix}$$ ( $B_n$ は使われない)

とするとき $S_n=\frac{P}{Q}$

**証明**

$$S_n=\frac{1}{C_0}(A_0+\frac{B_0}{C_1}(A_1+\frac{B_1}{C_2}(A_2+\frac{B_2}{C_3}(A_3+\cdots(A_{n-1}+\frac{B_{n-1}}{C_n}(A_n))))))$$

$$=\frac{1}{C_0}(A_0+B_0\cdot\frac{1}{C_1}(A_1+B_1\cdot\frac{1}{C_2}(A_2+B_2\cdot\frac{1}{C_3}(A_3+\cdots(A_{n-1}+B_{n-1}\cdot\frac{1}{C_n}(A_n+B_n\cdot 0))))))$$

これを関数 $f_i(x)=\frac{1}{C_i}(A_i+B_i x)$ の合成関数を用いて表すと

$$S_n=f_0(f_1(f_2(f_3(\cdots f_n(0)))))$$

ここで

$$f_i(x)=\frac{A_i+B_i x}{C_i}=\frac{B_i x+A_i}{0x+C_i}$$

と変形できるから

$$\begin{pmatrix} B_0 & A_0 \\ 0 & C_0 \end{pmatrix}\begin{pmatrix} B_1 & A_1 \\ 0 & C_1 \end{pmatrix}\begin{pmatrix} B_2 & A_2 \\ 0 & C_2 \end{pmatrix}\cdots\begin{pmatrix} B_n & A_n \\ 0 & C_n \end{pmatrix}=\begin{pmatrix} T & P \\ 0 & Q \end{pmatrix}$$

とすると，定理 2 により

$$f_0(f_1(f_2(f_3(\cdots f_n(x)))))=\frac{Tx+P}{0\cdot x+Q}$$

よって

$$S_n=\frac{T\cdot 0+P}{0\cdot 0+Q}=\frac{P}{Q}$$

**証明終**

なお， $k\neq 0$ とするとき

$\begin{pmatrix} B_i & A_i \\ 0 & C_i \end{pmatrix}\begin{pmatrix} B_{i+1} & A_{i+1} \\ 0 & C_{i+1} \end{pmatrix}$ を $\begin{pmatrix} kB_i & A_i \\ 0 & C_i \end{pmatrix}\begin{pmatrix} B_{i+1} & A_{i+1} \\ 0 & kC_{i+1} \end{pmatrix}$ に

$\begin{pmatrix} B_i & A_i \\ 0 & C_i \end{pmatrix}$ を $\begin{pmatrix} kB_i & kA_i \\ 0 & kC_i \end{pmatrix}$ に替えることができる。

例

$$e^a\approx 1+a+\frac{a^2}{2!}+\frac{a^3}{3!}+\cdots+\frac{a^n}{n!}$$

$$=1+\frac{a}{1}\left(1+\frac{a}{2}\left(1+\frac{a}{3}\left(1+\cdots\left(1+\frac{a}{n}\right)\right)\right)\right)$$

$$=1+a\cdot\frac{1}{1}\left(1+a\cdot\frac{1}{2}\left(1+a\cdot\frac{1}{3}\left(1+\cdots\left(1+a\cdot\frac{1}{n}(1+a\cdot 0)\right)\right)\right)\right)$$

$p$ を整数，$q$ を自然数として $a=\frac{p}{q}$ とすると

$$e^{\frac{p}{q}} \approx 1+\frac{p}{q}\cdot\frac{1}{1}\left(1+\frac{p}{q}\cdot\frac{1}{2}\left(1+\frac{p}{q}\cdot\frac{1}{3}\left(1+\cdots\left(1+\frac{p}{q}\cdot\frac{1}{n}\left(1+\frac{p}{q}\cdot 0\right)\right)\cdots\right)\right)\right)$$

$$=\frac{1}{1}\left(1+p\cdot\frac{1}{q}\left(1+p\cdot\frac{1}{2q}\left(1+p\cdot\frac{1}{3q}\left(1+\cdots\left(1+p\cdot\frac{1}{nq}\left(1+p\cdot 0\right)\right)\cdots\right)\right)\right)\right)$$

したがって

$$f_0(x)=\frac{1}{1}(1+px)=\frac{1+px}{1}=\frac{px+1}{0x+1}$$

$$f_i(x)=\frac{1}{iq}(1+qx)=\frac{1+px}{iq}=\frac{px+1}{0x+iq} \quad (i=1,\ 2,\ 3,\ \cdots,n)$$

とすると，

$$e^{\frac{p}{q}} \approx f_0(f_1(f_2(f_3(\cdots f_n(0)\cdots))))$$

と表される。このとき

$$\begin{pmatrix} p & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} p & 1 \\ 0 & q \end{pmatrix}\begin{pmatrix} p & 1 \\ 0 & 2q \end{pmatrix}\begin{pmatrix} p & 1 \\ 0 & 3q \end{pmatrix}\cdots\begin{pmatrix} p & 1 \\ 0 & nq \end{pmatrix}=\begin{pmatrix} T & P \\ 0 & Q \end{pmatrix}$$

とし

$$f(x)=\frac{Tx+P}{0x+Q}$$

とすると

$$e^{\frac{p}{q}} \approx f(0)=\frac{P}{Q}$$

## 連分数

$$S=b_0+\cfrac{a_1}{b_1+\cfrac{a_2}{b_2+\cfrac{a_3}{\cfrac{\ddots}{b_{n-1}+\cfrac{a_n}{b_n+0}}}}}$$

を考える。

$$f_i(x)=\frac{a_i}{b_i+x}=\frac{0\cdot x+a_i}{1\cdot x+b_i}$$

とすると

$$S=b_0+f_1(f_2(f_3(\cdots f_n(0)\cdots)))$$

となるから

$$\begin{pmatrix}0 & a_1\\ 1 & b_1\end{pmatrix}\begin{pmatrix}0 & a_2\\ 1 & b_2\end{pmatrix}\begin{pmatrix}0 & a_3\\ 1 & b_3\end{pmatrix}\cdots\begin{pmatrix}0 & a_n\\ 1 & b_n\end{pmatrix}=\begin{pmatrix}T & P\\ U & Q\end{pmatrix}$$

とすると，

$$S=b_0+\frac{T\cdot 0+P}{U\cdot 0+Q}=b_0+\frac{P}{Q}$$

例

$$\pi=4\arctan 1\approx\cfrac{4}{1+\cfrac{1^2}{3+\cfrac{2^2}{\cfrac{\ddots}{2n-1+\cfrac{n^2}{2n+1+0}}}}}$$

であるから

$$\begin{pmatrix}0 & 4\\ 1 & 1\end{pmatrix}\begin{pmatrix}0 & 1^2\\ 1 & 3\end{pmatrix}\begin{pmatrix}0 & 2^2\\ 1 & 5\end{pmatrix}\cdots\begin{pmatrix}0 & n^2\\ 1 & 2n+1\end{pmatrix}=\begin{pmatrix}T & P\\ U & Q\end{pmatrix}$$

とすると，

$$\pi\approx\frac{T\cdot 0+P}{U\cdot 0+Q}=\frac{P}{Q}$$

## 基数変換

例えば，r 進数を十進数表現に直す方法を考える。

$a$ の $r$ 進数表現は

$$a=a_n r^n+a_{n-1}r^{n-1}+a_{n-2}r^{n-2}+\cdots+a_0$$
$$=(\cdots((((0\cdot r+a_n)r+a_{n-1})r+a_{n-2})r+a_{n-3})r+\cdots)+a_0$$

ここで

$$f_i(x)=xr+a_i=\frac{r\cdot x+a_i}{0\cdot x+1}$$

とすると

$$a=f_0(f_1(f_2(f_3(\cdots f_n(0)\cdots))))$$

となるから

$$\begin{pmatrix}r & a_0\\ 0 & 1\end{pmatrix}\begin{pmatrix}r & a_1\\ 0 & 1\end{pmatrix}\begin{pmatrix}r & a_2\\ 0 & 1\end{pmatrix}\cdots\begin{pmatrix}r & a_n\\ 0 & 1\end{pmatrix}=\begin{pmatrix}T & P\\ 0 & Q\end{pmatrix}$$

とすると，

$$a = \frac{T \cdot 0 + P}{0 \cdot 0 + Q} = \frac{P}{Q}$$

この計算を十進数表現の多倍長演算プログラムで計算すればよい。

注　多項式で表された関数 $f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_0$ の値も同様に計算できる。

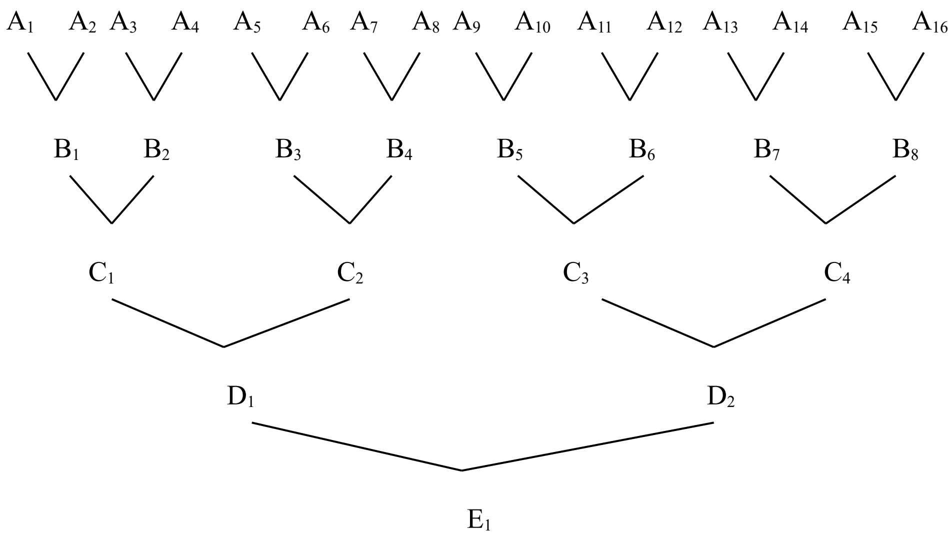
**BSA 法とその計算量**

掛ける行列の個数を n とする。まず，m を自然数として， $n = 2^m$ と表される場合を考える。

行列については，結合法則が成り立つから，1,2 番目，3,4 番目，5,6 番目…のように，隣り合う 2 個の

行列を掛けてもよい。この操作を行うたびに，掛けるべき行列の数は $\frac{1}{2}$ となる。（トーナメント法）

項の数が偶数でないときは，真ん中の項までとそれより後の項に分ければよい。




このとき，順に掛ける場合と乗算の回数は変わらないが，同じ程度の大きさの整数を掛けるため，高速フーリエ変換が効率的に使え，全体の計算量は小さくなる。また，並列処理ができる。

## 特別な漸化式で表される数列の和

**定理 3**　数列 $\{a_n\}$ が $c_n$，$s_n$ を整数， $t_n$ を自然数として

$$a_n = b_n c_n \quad (n = 0,\ 1,\ 2, \cdots, n)$$

$$b_0 = \frac{p}{q} \quad , \quad b_{n+1} = \frac{s_{n+1}}{t_{n+1}} \cdot b_n$$

すなわち

$$a_n=\frac{p}{q}\cdot\frac{s_1}{t_1}\cdot\frac{s_2}{t_2}\cdot\cdots\cdot\frac{s_n}{t_n}\cdot c_n \quad (n=0,\ 1,\ 2,\cdots,n)$$

の形に表されるとする。このとき

$$\begin{pmatrix} ps_1 & pc_0 \\ 0 & q \end{pmatrix}\begin{pmatrix} s_2 & c_1 \\ 0 & t_1 \end{pmatrix}\begin{pmatrix} s_3 & c_2 \\ 0 & t_2 \end{pmatrix}\cdots\cdots\begin{pmatrix} s_{n+1} & c_n \\ 0 & t_n \end{pmatrix}=\begin{pmatrix} T & P \\ 0 & Q \end{pmatrix}$$

とすると

$$\sum_{i=0}^{n} a_i=\frac{P}{Q}$$

**証明**

$$b_n=b_0\cdot\frac{s_1}{t_1}\cdot\frac{s_2}{t_2}\cdot\cdots\cdot\frac{s_n}{t_n}=\frac{p}{q}\cdot\frac{s_1}{t_1}\cdot\frac{s_2}{t_2}\cdot\cdots\cdot\frac{s_n}{t_n}$$

であるから

$$a_n=b_n\cdot c_n=\frac{p}{q}\cdot\frac{s_1}{t_1}\cdot\frac{s_2}{t_2}\cdot\cdots\cdot\frac{s_n}{t_n}\cdot c_n$$

よって $S_n=\sum_{i=0}^{n} a_i$ とすると

$$S_n=\frac{p}{q}\cdot c_0+\frac{p}{q}\cdot\frac{s_1}{t_1}\cdot c_1+\frac{p}{q}\cdot\frac{s_1}{t_1}\cdot\frac{s_2}{t_2}\cdot c_2+\cdots\cdots+\frac{p}{q}\cdot\frac{s_1}{t_1}\cdot\frac{s_2}{t_2}\cdot\cdots\cdot\frac{s_n}{t_n}\cdot c_n$$

$$=\frac{p}{q}\left(c_0+\frac{s_1}{t_1}\left(c_1+\frac{s_2}{t_2}\left(c_2+\cdots+\frac{s_{n-1}}{t_{n-1}}\left(c_{n-1}+\frac{s_n}{t_n}\cdot c_n\right)\cdots\right)\right)\right)$$

$$=\frac{1}{q}\left(pc_0+\frac{ps_1}{t_1}\left(c_1+\frac{s_2}{t_2}\left(c_1+\cdots+\frac{s_{n-1}}{t_{n-1}}\left(c_{n-1}+\frac{s_n}{t_n}\cdot c_n\right)\cdots\right)\right)\right)$$

したがって，定理 2 により

$$\begin{pmatrix} ps_1 & pc_0 \\ 0 & q \end{pmatrix}\begin{pmatrix} s_2 & c_1 \\ 0 & t_1 \end{pmatrix}\begin{pmatrix} s_3 & c_2 \\ 0 & t_2 \end{pmatrix}\cdots\cdots\begin{pmatrix} s_{n+1} & c_n \\ 0 & t_n \end{pmatrix}=\begin{pmatrix} T & P \\ 0 & Q \end{pmatrix}$$

とすると　$S_n=\frac{P}{Q}$

**証明終**

なお，　$P,\ Q$　の値は次のように求めてもよい。

$$\begin{pmatrix} s_2 & c_1 \\ 0 & t_1 \end{pmatrix}\begin{pmatrix} s_3 & c_2 \\ 0 & t_2 \end{pmatrix}\cdots\cdots\begin{pmatrix} s_{n+1} & c_n \\ 0 & t_n \end{pmatrix}=\begin{pmatrix} C & A \\ 0 & B \end{pmatrix}$$

とするとき

$$\begin{pmatrix} ps_1 & pc_0 \\ 0 & q \end{pmatrix} \begin{pmatrix} C & A \\ 0 & B \end{pmatrix} = \begin{pmatrix} T & P \\ 0 & Q \end{pmatrix}$$

例　**Chudnovsky の公式**

$$\frac{1}{\pi} = \frac{1}{426880\sqrt{10005}} \sum_{n=0}^{\infty} \frac{(-1)^n (6n)!\,(13591409 + 545140134n)}{(n!)^3 (3n)!\,640320^{3n}}$$

に BSA 法を適用して円周率を求めてみよう。

$$\frac{1}{426880} \sum_{n=0}^{\infty} \frac{(-1)^n (6n)!\,(13591409 + 545140134n)}{(n!)^3 (3n)!\,640320^{3n}}$$ において

$$a_n = \frac{(-1)^n (6n)!\,(13591409 + 545140134n)}{426880 (n!)^3 (3n)!\,640320^{3n}}$$

$$b_n = \frac{(-1)^n (6n)!}{426880 (n!)^3 (3n)!\,640320^{3n}} \quad , \quad c_n = 13591409 + 545140134n$$

とおくと　$b_0 = \dfrac{1}{426880}$　，　$a_n = b_n \cdot c_n$　であり

$$\frac{b_{n+1}}{b_n} = \frac{-(6n+1)(6n+2)(6n+3)(6n+4)(6n+5)(6n+6)}{(n+1)^3 (3n+1)(3n+2)(3n+3) 640320^3}$$

$$= \frac{-2\cdot 3\cdot 2\cdot 2(6n+1)(3n+1)(2n+1)(3n+2)(6n+5)(3n+3)}{(n+1)^3 (3n+1)(3n+2)(3n+3) 640320^3}$$

$$= \frac{-(2n+1)(6n+1)(6n+5)}{10939058860032000 (n+1)^3}$$

よって

$p = 1$，$q = 426880$

$c_i = 13591409 + 545140134 i \quad (i \geqq 0)$

$s_{i+1} = -(2i+1)(6i+1)(6i+5)$，$t_{i+1} = 10939058860032000 (i+1)^3 \quad (i \geqq 0)$

したがって

$$\begin{pmatrix} ps_1 & pc_0 \\ 0 & q \end{pmatrix} = \begin{pmatrix} -(2\cdot 0+1)(6\cdot 0+1)(6\cdot 0+5) & 13591409 + 545140134\cdot 0 \\ 0 & 426880 \end{pmatrix}$$

$i \geqq 1$ のとき

$$\begin{pmatrix} s_{i+1} & c_i \\ 0 & t_i \end{pmatrix} = \begin{pmatrix} -(2i+1)(6i+1)(6i+5) & 13591409 + 545140134 i \\ 0 & 10939058860032000 i^3 \end{pmatrix}$$

このとき

$$\pi \approx \sqrt{10005} \cdot \frac{Q}{P}$$

ここで

$$\lim_{n \to \infty} \frac{a_{n+1}}{a_n} = \lim_{n \to \infty} \frac{b_{n+1}}{b_n} = \frac{1}{151931373056000}$$

$$\log_{10}(151931373056000) = 14.1816\cdots$$

であるから，$n$ が 1 増えるごとに 14 桁ほど精度が増す。

十進 BASIC の 1000 桁モードで，行列の積をトーナメント方式で計算し，円周率を求めるプログラムは次のようになる。桁数が大きくなっても使えるように，再帰的なプログラムにした。

```
! Chudnovsky の公式  Binary Splitting
OPTION ARITHMETIC DECIMAL_HIGH
LET n=51 ! n 個の和
CALL BSA(T,P,Q,0,n-1)
LET pai=SQR(10005)*Q/P
PRINT pai-PI
END

EXTERNAL SUB BSA(T,P,Q,left,right)
OPTION ARITHMETIC DECIMAL_HIGH
IF right-left>0 THEN
   LET middle=INT((left+right)/2)
   CALL BSA(T0,P0,Q0,left,MIDDLE)
   CALL BSA(T1,P1,Q1,middle+1,right)
   LET T=T0*T1
   LET P=T0*P1+P0*Q1
   LET Q=Q0*Q1
ELSE
   LET T=-(2*left+1)*(6*left+1)*(6*left+5)
```

```
    LET P=13591409+545140134*left
    IF left=0 THEN
        LET Q=426880
    ELSE
        LET Q=10939058860032000*left^3
    END IF
END IF
END SUB
```

上記の十進 BASIC のプログラムを The GNU Multiple Precision Arithmetic Library (GMP)を使って C に移植したところ，8GB のメモリでで高速に 10 億桁が計算できました。ただし，10 億桁目は 10 億 1 桁目が四捨五入され、かつ，末尾の 0 は出力されないようです。
さらに，Ramanujan の公式，arctan 公式のプログラムを作成し，SHA-256 によってこれらの出力ファイルのハッシュ値が一致することを確かめました。

```
#include <stdio.h>
#include <stdlib.h>
#include <gmp.h>
#include <math.h>
#include <time.h>

void BSA(mpz_t T,mpz_t P,mpz_t Q,long left,long right,long end) {
    if (right>left){
        long middle=(left+right)/2;
        mpz_t T1,P1,Q1;
        mpz_init(T1);
        mpz_init(P1);
        mpz_init(Q1);
        BSA(T,P,Q,left,middle,end);
        BSA(T1,P1,Q1,middle+1,right,end);
```

```
        mpz_mul(P,P,Q1);
        mpz_addmul(P,T,P1);
        mpz_clear(P1);
        if (right==end) {
            mpz_clear(T);
            mpz_init(T);
        }else{
            mpz_mul(T,T,T1);
        }
        mpz_clear(T1);
        mpz_mul(Q,Q,Q1);
        mpz_clear(Q1);
    }else{
        mpz_set_si(T,-(2*left+1));
        mpz_mul_ui(T,T,6*left+1);
        mpz_mul_ui(T,T,6*left+5);
        mpz_set_ui(P,545140134);
        mpz_mul_ui(P,P,left);
        mpz_add_ui(P,P,13591409);
        if (left==0){
            mpz_set_ui(Q,426880);
        }else{
            mpz_set_str(Q,"10939058860032000",10);
            mpz_mul_ui(Q,Q,left);
            mpz_mul_ui(Q,Q,left);
            mpz_mul_ui(Q,Q,left);
        }
    }
}
```

```
int main () {
    FILE* fp;
    unsigned long digits=1000000000;
    unsigned long prec=(digits+5)*log2(10);
    long n=ceil(digits/14.18);
    mpf_t temp,pi;
    mpz_t T,P,Q;
    clock_t start=clock(),end;

    mpf_set_default_prec(prec);
    mpz_init(T);
    mpz_init(P);
    mpz_init(Q);
    BSA(T,P,Q,0,n,n);
    mpz_clear(T);
    mpf_init(pi);
    mpf_sqrt_ui(pi,10005);
    mpf_init(temp);
    mpf_set_z(temp,Q);
    mpz_clear(Q);
    mpf_mul(pi,pi,temp);
    mpf_set_z(temp,P);
    mpz_clear(P);
    mpf_div(pi,pi,temp);
    mpf_clear(temp);
    end=clock();
    printf("%.3f s\n",(double)(end-start)/CLOCKS_PER_SEC);

    fp=fopen("pi.txt","w");
    mpf_out_str(fp,10,digits+1,pi);
```

```
	fclose(fp);
	mpf_clear(pi);
	end=clock();
	printf("%.3f s\n",(double)(end-start)/CLOCKS_PER_SEC);
	return 0;
}
```

例　**Ramanujan の公式**

$$\frac{1}{\pi}=\frac{\sqrt{8}}{9801}\sum_{n=0}^{\infty}\frac{(4n)!(1103+26390n)}{(n!)^4\ \ 396^{4n}}$$

に BSA 法を適用して円周率を求めてみよう。

$$\frac{1}{9801}\sum_{n=0}^{\infty}\frac{(4n)!(1103+26390n)}{(n!)^4\ \ 396^{4n}}$$ において

$$a_n=\frac{(4n)!(1103+26390n)}{9801(n!)^4\ \ 396^{4n}}$$

$$b_n=\frac{(4n)!}{9801(n!)^4\ \ 396^{4n}}\quad,\quad c_n=1103+26390n$$

とおくと　$b_0=\frac{1}{9801}$　,　$a_n=b_n\cdot c_n$　であり

$$\frac{b_{n+1}}{b_n}=\frac{(4n+1)(4n+2)(4n+3)(4n+4)}{24591257856(n+1)^4}$$

$$=\frac{(2n+1)(4n+1)(4n+3)}{3073907232(n+1)^3}$$

よって

$$p=1,\ q=9801$$

$$c_i=1103+26390i\ \ (i\geqq 0)$$

$$s_{i+1}=(2i+1)(4i+1)(4i+3),\ t_{i+1}=3073907232(i+1)^3\ \ (i\geqq 0)$$

したがって

$$\begin{pmatrix} ps_1 & pc_0 \\ 0 & q \end{pmatrix}=\begin{pmatrix} (2\cdot 0+1)(4\cdot 0+1)(4\cdot 0+3) & 1103+26390\cdot 0 \\ 0 & 9801 \end{pmatrix}$$

$i\geqq 1$ のとき

$$\begin{pmatrix} s_{i+1} & c_i \\ 0 & t_i \end{pmatrix} = \begin{pmatrix} (2i+1)(4i+1)(4i+3) & 1103+26390i \\ 0 & 3073907232i^3 \end{pmatrix}$$

このとき

$$\pi \approx \frac{Q}{P \cdot \sqrt{8}}$$

ここで

$$\lim_{n \to \infty} \frac{a_{n+1}}{a_n} = \lim_{n \to \infty} \frac{b_{n+1}}{b_n} = \frac{1}{96059601}$$

$$\log_{10}(96059601) = 7.98254\cdots$$

であるから，$n$ が 1 増えるごとに 8 桁近く精度が増す。

**例　arctan 公式による円周率の導出**

$k$，$m$ を自然数として，$4k \arctan \frac{1}{m}$ テーラー展開し，定理 3 を用いる。

$$4k \arctan \frac{1}{m} = \sum_{n=1}^{\infty} \frac{4k(-1)^{n-1}}{2n-1} \cdot \frac{1}{m^{2n-1}} = \sum_{n=0}^{\infty} \frac{4k(-1)^n}{2n+1} \cdot \frac{1}{m^{2n+1}}$$

であるから

$$a_n = b_n = \frac{4k(-1)^n}{2n+1} \cdot \frac{1}{m^{2n+1}} \quad , \quad c_n = 1 \quad , \quad b_0 = \frac{4k}{m}$$

$$p = 4k, \ q = m$$

$$\frac{b_{n+1}}{b_n} = -\frac{2n+1}{(2n+3)m^2} \quad \text{より} \quad s_{i+1} = -(2i+1), \ t_{i+1} = (2i+3)m^2$$

よって

$$\begin{pmatrix} p s_1 & p c_0 \\ 0 & q \end{pmatrix} = \begin{pmatrix} -4k & 4k \\ 0 & m \end{pmatrix}$$

$i \geqq 1$ のとき

$$\begin{pmatrix} s_{i+1} & c_i \\ 0 & t_i \end{pmatrix} = \begin{pmatrix} -(2i+1) & 1 \\ 0 & (2i+1)m^2 \end{pmatrix}$$

したがって，次の公式の十進 BASIC によるプログラムは下のようになる。

$$\pi = 4*(83 \arctan \frac{1}{107} + 17 \arctan \frac{1}{1710} - 44 \arctan \frac{1}{225443}$$

$$-68\arctan\frac{1}{2513489}+22\arctan\frac{1}{42483057}+34\arctan\frac{1}{7939642926390344818})$$

```
! arctanの公式 Binary Splitting
OPTION ARITHMETIC DECIMAL_HIGH
LET pai=arctan(83,107,2^7)+arctan(17,1710,2^6)-arctan(44,225443,2^6)-arctan(68,2513489,2^6)+arctan(22,42483057,2^5)+arctan(34,7939642926390344818,2^4)
PRINT pai-PI
END

EXTERNAL FUNCTION arctan(p,q,n)
OPTION ARITHMETIC DECIMAL_HIGH
CALL BSA(A,B,C,p,q,q^2,0,n)
LET arctan=A/B
END FUNCTION

EXTERNAL SUB BSA(A,B,C,p,q,q2,left,right)
OPTION ARITHMETIC DECIMAL_HIGH
IF right>left THEN
  LET mid=INT((left+right)/2)
  CALL BSA(A0,B0,C0,p,q,q2,left,mid)
  CALL BSA(A1,B1,C1,p,q,q2,mid+1,right)
  LET C=C0*C1
  LET A=C0*A1+A0*B1
  LET B=B0*B1
ELSE
  IF left=0 THEN
    LET C=-4*p
    LET A=4*p
    LET B=q
```

```
  ELSE
    LET C=-2*left-1
    LET A=1
    LET B=(2*left+1)*q2
  END IF
END IF
END SUB
```

**Ramanujanの公式による計算法のCとGMPへの移植**

Ramanujanの公式によるプログラムをBinary Splitting Algorithmを用いてGMPを使ってCで書いたところ，8MBのメモリではメモリがギリギリの状態で10億桁を計算できました。

```
#include <stdio.h>
#include <stdlib.h>
#include <gmp.h>
#include <math.h>
#include <time.h>

void BSA(mpz_t T,mpz_t P,mpz_t Q,long left,long right,long end) {
    if (right>left){
        long middle=(left+right)/2;
        mpz_t T1,P1,Q1;
        mpz_init(T1);
        mpz_init(P1);
        mpz_init(Q1);
        BSA(T,P,Q,left,middle,end);
        BSA(T1,P1,Q1,middle+1,right,end);
        mpz_mul(P,P,Q1);
        mpz_addmul(P,T,P1);
        mpz_clear(P1);
```

```
        if (right==end) {
            mpz_clear(T);
            mpz_init(T);
        }else{
            mpz_mul(T,T,T1);
        }
        mpz_clear(T1);
        mpz_mul(Q,Q,Q1);
        mpz_clear(Q1);
    }else{
        mpz_set_ui(T,2*left+1);
        mpz_mul_ui(T,T,4*left+1);
        mpz_mul_ui(T,T,4*left+3);
        mpz_set_ui(P,26390);
        mpz_mul_ui(P,P,left);
        mpz_add_ui(P,P,1103);
        if (left==0){
            mpz_set_ui(Q,9801);
        }else{
            mpz_set_ui(Q,3073907232);
            mpz_mul_ui(Q,Q,left);
            mpz_mul_ui(Q,Q,left);
            mpz_mul_ui(Q,Q,left);
        }
    }
}

int main () {
    FILE* fp;
    unsigned long digits=1000000000;
```

```
unsigned long prec=(digits+5)*log2(10);
long n=ceil(digits/7.98);
mpf_t temp,pi;
mpz_t T,P,Q;
clock_t start=clock(),end;

mpf_set_default_prec(prec);
mpz_init(T);
mpz_init(P);
mpz_init(Q);
BSA(T,P,Q,0,n,n);
mpz_clear(T);
mpf_init(pi);
mpf_sqrt_ui(pi,8);
mpf_init(temp);
mpf_set_z(temp,P);
mpz_clear(P);
mpf_mul(pi,pi,temp);
mpf_set_z(temp,Q);
mpz_clear(Q);
mpf_div(pi,temp,pi);
mpf_clear(temp);
end=clock();
printf("%.3f s\n",(double)(end-start)/CLOCKS_PER_SEC);

fp=fopen("pi.txt","w");
mpf_out_str(fp,10,digits+1,pi);
fclose(fp);
mpf_clear(pi);
end=clock();
```

```
    printf("%.3f s\n",(double)(end-start)/CLOCKS_PER_SEC);
    return 0;
}
```

**逆正接(arctan)公式による計算法のCとGMPへの移植**

逆正接公式のプログラムをBinary Splitting Algorithmを用いてGMPを使ってCで書いたところ,

8GBのメモリで10億桁が計算できました。

```
#include <stdio.h>
#include <stdlib.h>
#include <gmp.h>
#include <math.h>
#include <time.h>

void BSA(mpz_t T,mpz_t P,mpz_t Q,long p,mpz_t QZ,mpz_t QQ,long left,long right,long end) {
  if (right>left){
        long middle=(left+right)/2;
        mpz_t T1,P1,Q1;
        mpz_init(T1);
        mpz_init(P1);
        mpz_init(Q1);
        BSA(T,P,Q,p,QZ,QQ,left,middle,end);
        BSA(T1,P1,Q1,p,QZ,QQ,middle+1,right,end);
        mpz_mul(P,P,Q1);
        mpz_addmul(P,T,P1);
        mpz_clear(P1);
        if (right==end) {
            mpz_clear(T);
            mpz_init(T);
        }else{
```

```
                mpz_mul(T,T,T1);
            }
            mpz_clear(T1);
            mpz_mul(Q,Q,Q1);
            mpz_clear(Q1);
    }else{
      if (left==0){
            mpz_set_si(T,-4*p);
            mpz_set_si(P,4*p);
            mpz_set(Q,QZ);
            }else{
            mpz_set_si(T,-2*left-1);
            mpz_set_ui(P,1);
            mpz_mul_ui(Q,QQ,2*left+1);
            }
    }
}

void ARCTAN0(mpf_t arctan,long p,char *q,unsigned long digits){
      mpz_t P,Q,T,QZ,QQ;
      mpf_t temp;
      mpz_init(QZ);
      mpz_init(QQ);
      mpz_set_str(QZ,q,10);
      mpz_mul(QQ,QZ,QZ);
      long exp;
      double a=mpz_get_d_2exp(&exp,QQ);
      long n=digits/((exp+log2(a))*log10(2));
      mpz_init(P);
      mpz_init(Q);
```

```
    mpz_init(T);
    BSA(T,P,Q,p,QZ,QQ,0,n,n);
    mpz_clear(QZ);
    mpz_clear(QQ);
    mpz_clear(T);
    mpf_init(arctan);
    mpf_set_z(arctan,P);
    mpz_clear(P);
    mpf_init(temp);
    mpf_set_z(temp,Q);
    mpz_clear(Q);
    mpf_div(arctan,arctan,temp);
    mpf_clear(temp);
}

void ARCTAN(mpf_t pi,long p,char *q,unsigned long digits){
    mpf_t arctan;
    ARCTAN0(arctan,p,q,digits);
    mpf_add(pi,pi,arctan);
    mpf_clear(arctan);
}

void TIME(clock_t start){
    clock_t end=clock();
    printf("%.3f s\n",(double)(end-start)/CLOCKS_PER_SEC);
}

int main () {
    FILE* fp;
    unsigned long digits=1000000000;
```

```
	unsigned long prec=(digits+5)*log2(10);
	mpf_t pi;
	clock_t start=clock();

	mpf_set_default_prec(prec);
	ARCTAN0(pi,83,"107",digits);
	TIME(start);
	ARCTAN(pi,17,"1710",digits);
	TIME(start);
	ARCTAN(pi,-44,"225443",digits);
	TIME(start);
	ARCTAN(pi,-68,"2513489",digits);
	TIME(start);
	ARCTAN(pi,22,"42483057",digits);
	TIME(start);
	ARCTAN(pi,34,"793964292639034481 8",digits);
	TIME(start);

	fp=fopen("pi.txt","w");
	mpf_out_str(fp,10,digits+1,pi);
	fclose(fp);
	mpf_clear(pi);
	TIME(start);
	return 0;
}
```

例　mを自然数として　$1-\cos\frac{1}{\sqrt{m}}$　のテーラー展開

$$1-\cos\frac{1}{\sqrt{m}}=\sum_{n=1}^{\infty}\frac{(-1)^{n-1}}{(2n)!}\cdot\frac{1}{\left(\sqrt{m}\right)^{2n}}=\sum_{n=0}^{\infty}\frac{(-1)^{n}}{(2n+2)!}\cdot\frac{1}{m^{n+1}}$$

であるから

$$a_n=b_n=\frac{(-1)^n}{(2n+2)!m^{n+1}}\quad,\quad c_n=1\quad,\quad b_0=\frac{1}{2m}$$

$p=1$，$q=2m$

$$\frac{b_{n+1}}{b_n}=\frac{-1}{(2n+3)(2n+4)m}$$ より $s_{i+1}=-1$，$t_{i+1}=(2i+3)(2i+4)m$

よって

$$\begin{pmatrix} ps_1 & pc_0 \\ 0 & q \end{pmatrix}=\begin{pmatrix} -1 & 1 \\ 0 & 2m \end{pmatrix}\equiv\begin{pmatrix} 1 & -1 \\ 0 & -2m \end{pmatrix}$$

$i\geqq 1$ のとき

$$\begin{pmatrix} s_{i+1} & c_i \\ 0 & t_i \end{pmatrix}=\begin{pmatrix} -1 & 1 \\ 0 & (2i+1)(2i+2)m \end{pmatrix}\equiv\begin{pmatrix} 1 & -1 \\ 0 & -(2i+1)(2i+2)m \end{pmatrix}$$

まとめると，$i\geqq 0$ のとき

$$\begin{pmatrix} s_{i+1} & c_i \\ 0 & t_i \end{pmatrix}=\begin{pmatrix} 1 & -1 \\ 0 & -(2i+1)(2i+2)m \end{pmatrix}$$

## BSA 法の計算量についての考察

掛ける行列の数は $n$ 個とし，各行列の成分は $b$ ビットの整数とする。$b$ ビット同士の積の計算量を M($b$)

とし， $n=2^m$ すなわち $m=\log_2 n$ とすると，トーナメントは $m$ 段の積となる。総計算量 T は

$$T=M(b)\cdot\frac{n}{2}+M(2b)\cdot\frac{n}{4}+M(4b)\cdot\frac{n}{8}+\cdots+M(2^{m-1}b)\cdot 1$$

$M(b)=cb^2$ のとき

$$T=c\{b^2\cdot\frac{n}{2}+(2b)^2\cdot\frac{n}{4}+(4b)^2\cdot\frac{n}{8}+\cdots+(2^{m-1}b)^2\cdot 1\}$$

$$=\frac{cb^2n}{2}\cdot(1+2+4+\cdots+2^{m-1})=\frac{cb^2n}{2}\cdot\frac{2^m-1}{2-1}$$

$$\approx\frac{cb^2n}{2}\cdot 2^m=\frac{cb^2}{2}\cdot n^2=\frac{M(b)}{2}\cdot n^2$$

しかし，高速フーリエ変換では $M(b)\approx cb\log_2 b$ であるから

$$T\approx c\{b\log_2 b\cdot\frac{n}{2}+2b\log_2(2b)\cdot\frac{n}{4}+4b\log_2(4b)\cdot\frac{n}{8}+\cdots+\frac{n}{2}\cdot b\log_2(2^{m-1}b)\cdot 1\}$$

$$=\frac{cbn}{2}\cdot\{\log_2 b+\log_2(2b)+\log_2(4b)+\log_2(2^{m-1}b)\}$$

$$=\frac{cbn}{2}\cdot\{m\log_2 b+1+2+3+(m-1)\}$$

$$=\frac{cbn}{2}\cdot\{m\log_2 b+\frac{(m-1)m}{2}\}$$

$$\approx\frac{cbn}{2}\cdot(m\log_2 b+\frac{m^2}{2})$$

$$\approx\frac{cbnm}{2}\cdot(\log_2 b+\frac{m}{2})$$

$$=\frac{cbn\log_2 n}{2}\cdot\left(\log_2 b+\frac{\log_2 n}{2}\right)$$

$$=\frac{1}{4}cbn(\log_2 n)^2\left(1+\frac{2\log_2 b}{\log_2 n}\right)$$

級数の和を，小数点以下$N$ビットまで求めるとし，級数の各項は$k$ビットずつ小さくなるとする。さらに，$p$を小さな自然数として，各行列の成分が　$2^k\times n^p$　程度である場合を考える。このとき

$$n\approx\frac{N}{k}\quad,\quad b\approx\log_2(2^k n^p)$$

である。まとめると

$$T\approx\frac{1}{4}cbn(\log_2 n)^2\left(1+\frac{2\log_2 b}{\log_2 n}\right)\quad,\quad n\approx\frac{N}{k}\quad,\quad b\approx k+p\log_2 N$$

特に　$\log_2 N\gg\log_2 k$　のときの近似を考える。

$$\log_2 n\approx\log_2\frac{N}{k}=\log_2 N-\log_2 k\approx\log_2 N$$

$$b\approx k+p\log_2 n\approx k+p\log_2 N$$

ここで，　$k>2$，$p\log_2 N>2$　であるから

$$\log_2(k+p\log_2 N)<\log_2 k+\log_2(p\log_2 N)\ll\log_2 N$$

よって

$$\log_2 b\approx\log_2(k+p\log_2 N)\ll\log_2 N\approx\log_2 n\quad\text{すなわち}\quad\frac{\log_2 b}{\log_2 n}\ll 1$$

したがって

$$T\approx\frac{1}{4}cbn(\log_2 n)^2$$

$b$，$n$　を消去して　$k$，$p$，$N$　で表すと

$$T \approx \frac{1}{4} c\left(k + p \log_2 N\right) \cdot \frac{N}{k} \cdot \left(\log_2 N\right)^2 = \frac{1}{4} c N \left(\log_2 N\right)^2 \left(\frac{p}{k} \log_2 N + 1\right)$$

よって　$N \rightarrow \infty$　の極限では　$T \approx \frac{cp}{4k} N\left(\log_2 N\right)^3$　となる。しかし，円周率の世界記録は 50 兆桁で無限大では無いから，$k$ が大きいときに，$\frac{p}{k} \log_2 N + 1$ において $\frac{p}{k} \log_2 N$ は 1 と比べて十分大きくはない。

**例**　Størmer 公式

$$\frac{\pi}{4} = 44 \arctan\left(\frac{1}{57}\right) + 7 \arctan\left(\frac{1}{239}\right) - 12 \arctan\left(\frac{1}{682}\right) + 24 \arctan\left(\frac{1}{12943}\right)$$

で 50 兆桁を計算するとして，$\arctan\left(\frac{1}{57}\right)$　を計算するときを考える。

$$N = 5.0 \times 10^{13} \times \log_2 10 \quad \text{より} \quad \log_2 N \approx 47.24$$

行列は $\begin{pmatrix} -(2i+1) & 1 \\ 0 & (2i+1)(57)^2 \end{pmatrix}$ であり，　$(2i+1)(57)^2$　に注目すると

$$p = 1,\ k = \log_2 (57)^2 \approx 11.67$$

$\log_2 N \gg \log_2 k$　としてもよいであろう。

$$\frac{p}{k} \log_2 N \approx 4.05 \quad \text{であるから} \quad \frac{p}{k} \log_2 N + 1 \approx 5.05$$

また，$\arctan\left(\frac{1}{12943}\right)$　を計算するとき

$$p = 1,\ k = \log_2 (12943)^2 \approx 27.32$$

よって

$$\frac{p}{k} \log_2 N \approx 1.73 \quad \text{であるから} \quad \frac{p}{k} \log_2 N + 1 \approx 2.73$$

**例**　Chudnovsky の公式で 50 兆桁を計算するとき

行列は $\begin{pmatrix} -(2i+1)(6i+1)(6i+5) & 13591409 + 545140134 i \\ 0 & 10939058860032000 i^3 \end{pmatrix}$ であり，　$10939058860032000 i^3$　に注目すると

$$p = 3,\ k = \log_2 10939058860032000 \approx 53.28$$

よって

$\frac{p}{k}\log_2 N \approx 2.66$　であるから　$\frac{p}{k}\log_2 N + 1 \approx 3.66$