π 值估算 说明文档

1900012476 夏罗生

1 Monte Carlo 算法

在 $1*1$ 的矩形区域内随机撒点，在 $\frac{1}{4}$ 圆内的概率为 $\frac{\pi}{4}$。

Monte Carlo 算法的收敛速度很慢，时间复杂度约为 $O(10^N)$。

C++ 内置类型 double 为 53 位，在十进制下的最大精度位 15 位，能够满足算法的精度要求。

2 Leibniz 级数

利用反正切级数：

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \ldots$$

代入 $x = 1$，得到：

$$\pi = 4\sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

这个级数的收敛速率是次线性，100 项误差能精确到小数点后 2 位。double 类型同样能满足算法的精度要求

3 Chudnovsky 公式的变式

Chudnovsky 公式是曾经用来打破圆周率精度世界纪录的式子，它方便计算机编程的表达式为：

$$\frac{1}{\pi} = \frac{1}{53360\sqrt{640320}} \sum_{k=0}^{\infty} (-1)^k \frac{(6k)!(13591409 + 545140134k)}{(3k)!(k!)^3(640320)^{3k}}$$

线性收敛，每算一项得到 15 位十进制精度。

由于 Chudnovsky 公式的收敛速度太快，而 C++ 最大的浮点内置类型 long double 只有 64 位，在十进制下的最大精度只有 18 位，远远不能满足我们的需求。需要自定义高精度的浮点数 Big Float。

4 Big Float

4.1 数据结构

```cpp
private:
    char sign;
    std::deque<char> number;
    int decimals;
```

使用一个字符 `sign` 表示正负号，使用 `Vector` 容器中的每一项储存十进制下的每一位，使用一个整型 `decimals` 表示容器中小数部分的长度。

4.2 构造函数

构造函数只用实现 `BigFloat(const char* strNum)` 即可。

对于重载的其他构造函数，可以将参数转换为 `string` 类型，再转换为 `const char*` 类型，再调用 `BigFloat(const char* strNum)` 实现。

```
public:
    BigFloat();
    BigFloat(const char* strNum);
    BigFloat(std::string strNum);
    BigFloat(int Num);
    BigFloat(long long Num);
    BigFloat(double Num);
```

由于没有实现开根号，所以将 Chudnovsky 公式中的 $53360\sqrt{640320}$ 事先计算出来，使用字符串形式构造为 BigFloat。

```
string const_sqrt = "42698670.666333395817712889160659608273320884002509082800838007178852605157457594216301799911455668601345737167494080411392292736181266728193136882170582563460066798766483460795735983552333985484854583";
BigFloat(const_sqrt);
```

4.3 精度控制

自定义数据类型和核心就是设计加减乘除运算的结果精度，BigFloat 的精度设计和 C++ 由相似之处。

- 加法 减法 除法：取操作数中最多的小数位数为结果的小数位数。
- 乘法：取操作数中小数位数之和为结果的小数位数。

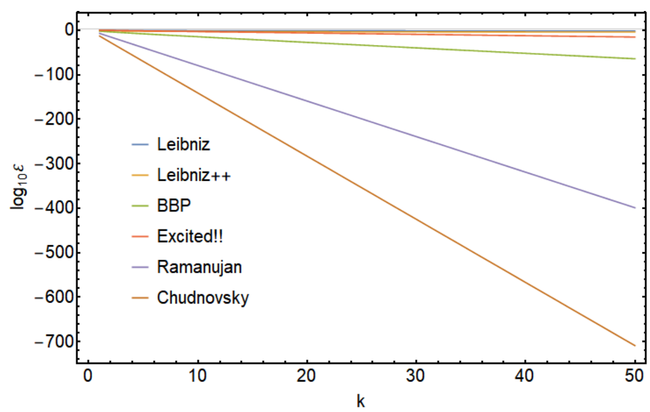
此时两个整数相除仍然是整数，所以要再增加一个函数：

- 自定义精度除法：将小数位数作为参数传入。

我们在计算 Chudnovsky 公式会使用到自定义精度除法，此时我们要提前确定需要的精度。

Chudnovsky 公式位线性收敛，每算一项得到 15 个有效数字，所以应估算 π 的小数位数 $d = 15n$。

同时考虑到中间项 $(-1)^k \frac{(6k)!(13591409+545140134k)}{(3k)!(k!)^3(640320)^{3k}}$ 很小，在小数部分中有很多 0，为了保证有足够得有效位数，中间项的小数位数 $d = 30n$。



4.4 计算顺序

对于 Chudnovsky 公式的每一项 $(-1)^k\frac{(6k)!(13591409+545140134k)}{(3k)!(k!)^3(640320)^{3k}}$。设计每一子项操作数的类型，尽量使用内置类型以减少计算量。

C++ 中最大的整型内置类型为 long long 为 64位，最多能表示 10^{18} 数量级的整数。

下表列出各子项大概的数量级：

表达式	k	**大小**	**理想的数据类型**
$\frac{(6k)!}{(3k)!}$	7	$2.8*10^{31}$	BigFloat
$(13591409++545140134k)$	10	$5.6*10^{8}$	BigFloat
$(k!)^3$	7	$1.3*10^{11}$	long long
$(640320)^{3k}$	1	$-2.6*10^{17}$	BigFloat

$\frac{(6k)!}{(3k)!}$ 使用一个 $(3k,6k]$ 范围的循环计算结果，结果用 BigFloat 类型表示。$(640320)^{3k}$ 写成 $(640320^3)^k$，括号内的乘方使用 long long 计算，括号外的乘方每次迭代计算一次。