Big Numbers

Basic info

基础功能

你需要实现一个大数类。

大数类的基类应当是一个二进制串，因此你需要一个基类用于存放所有需要的 bit 。使用二进制可以完全利用所有申请到的空间。你可以使用 STL 中的容器来存储这些 bit （即整数）。

你可以对二进制串定义一些基础操作，例如与、或、非、比较大小、添加 bit 等。当然，例如比较大小的功能可能在派生类中被复写。

大数类的派生类为一个整数类，需要实现整数的加法、减法、乘法和针对小整数的除法。

大数类、二进制串都应当有完善的构造和析构函数，两者都需要允许向一些常见的数据结构（例如 `std::string` 或 `std::vector` ）显式转换。

附加功能

- 实现大数对大数的除法。
- 对二进制串增添新的派生类浮点类，需要遵循一个浮点数的标准（可以参考 IEEE-754）

  浮点类需要重写若干基础操作
- 对二进制串和大数类增添迭代器
- 空间分配方面有两档附加功能
  - 可以通过模版类的形式让你的二进制串或大整数类对用户指定的 STL 容器进行包装。也就是说用户可以指定你的串会使用的 STL 容器类型。
  - 自己尝试实现空间的分配和释放，可以参考使用 `allocator` 。
- 实现流输入输出。即支持 `cin,cout`

  可以尝试额外支持 `HEX,OCT` 等参数。
- 学习算法加速大数乘法和除法