Boost.multiprecision

Massand Sagar Sunil
11414
sagarm@iitk.ac.in

24th April,2013

1 Personal details

- **Name:** Massand Sagar Sunil
- **Institute:** Indian Institute of Technology,Kanpur
- **Course:** Computer Science and Engineering
- **Degree program:** B.Tech
- **Email:** sagar.massand@gmail.com
- **Availability:**
  I am not involved in any major activity (courses/internships) from June to mid-August.However,I would be able to work for only around 30 hours a week from mid-August to September. Hence, I intend to work harder in the first half of the project (about 50 hours a week),starting from about June 10th and then slow down towards the end.

2 Background Information

I am currently pursuing my B.Tech in Computer Science and Engineering from IIT-Kanpur.I have long felt the need for a bigint class in C++.I believe that the language would be significantly better for users with the addition of a well-optimized bigint class.I am very interested in algorithms and their implementation and want to improve my coding skills to be able to participate as a serious contender in competitions like the ACM-ICPC, Google Code Jam etc. I believe that this project would help me in improving my coding greatly.
As a part of a project, I had done a naive implementation of Karatsuba for a radix 2 multiplication (which is attached) and had been fascinated by the complexities of writing a code which frequently exceeded memory limits due to the recursive calls. I had also done an implementation of a gaussian random number generator in verilog HDL using the central limit theorem which also used an implentation of the Goldschmidt algorithm.
I have a decent knowledge of C++ as well as the STL library. I haven't worked on subversion however, I have worked on github before and have a fair knowledge about it. I am fairly familiar with Doxygen for documentation.

3 Proposed implementation

My primary focus in the project (which I believe would take up a lot of time) would be the following:

- Implementation of an efficient Karatsuba algorithm with all the required testing for it as my primary aim is to make it workable for uptil 10,000 digits.I am in favour of going for Karatsuba as it has been mentioned that we are going for radix 2 implementation and my own naive code was more than 2.5 times faster than the normal multiplication algorithm for an input size of just a 1000 bits.
- Researching on certain approximation division algorithms like the Newton-Raphson,Goldschmidt algorithm etc.,their implementation and appropriate testing for these algorithms. I also intend to add a parameter for division which can be optionally given to specify the precision.

4 Proposed milestones and schedule

- **June 10-17:** Implementation of basic addition, subtraction and Karatsuba algorithm.Further optimizations can be done later but a basic code should be done in this interval.
- **June 17-July 6:** Research on the performance of division algorithms like the Newton-Raphson,Goldschmidt's algorithm etc., their basic implementation, possible optimizations etc. along with the writing of different tests for these algorithms.
- **July 7-18:** Research on the Schonhage-Strassen algorithm as well as other algorithms based on Fast-Fourier transforms.
- **July 19-August 5:** Implementation of a Fast-Fourier transform algorithm according to the research done.
- **August 6-August 15:** Generation of tests for the multiplication algorithms.
- **August 16-August 28:** Implementation of transcendental functions like exponential function, logarithm function etc.
- **August 29-September 10:** Buffer period for any delays in the above, or some last minute feature to be added etc.