

GSOC'14 PROPOSAL

Personal Details

Name: Saurabh Jain

Country: India

University/College: Indian Institute of Technology Mandi (IIT - Mandi), Himachal Pradesh, India
– <http://iitmandi.ac.in>

Years completed: 3rd year

Current program: B.Tech in Computer Science and Engineering and a junior.

Email: c@saurabhj.net

Phone: +91-9805377265, +91-9717563717

Other contact channels: gchat – saurabhgggn.94, skype – saurabhskj

Personal website: <http://saurabhj.net>

Project Description

Project Title: Search Trees and Priority Queues

Detailed Description: Following the brief description given for this project on ideas page, the aim of this project will be first to implement data structures like Priority queue, balanced BST, splay tree, skip list, fibonacci heaps and many others in C++ 11 and also to implement various algorithms like Dijkstra's algo using different data structures like F. Heaps, Priority queue, etc. So as to compare the time bounds and use the algorithm more efficiently depending on the data sets we have.

What have you done so far?: I have implemented most of the algorithms mentioned above in C++ (others in Java/ Python) and now creating templates for the classes in each of them so as to have multiple implementations of same algorithm using different data structures.

How do you plan to realize your project?: Project will be divided into many sub projects and the task for each sub-project will be accomplished in a week before moving onto next sub-project. At the beginning of each week, first the requirements analysis for that sub-project will be done followed by its design architecture so that high level optimization can be done initially itself. Then, the rest of the week will focus on implementing the features as mentioned in the requirement phase followed by final testing.

The following task will be accomplished in the subsequent weeks:

Priority Queues: Implementation using following data structures - Binary heaps, Fibonacci heaps, Binomial heaps, Pairing heaps, Heap-On-Top queues, *Associative containers – map and *Balanced Search trees.

Dictionary: Balanced Search trees- AVL & Red-Black, hashing, skiplists, Balanced Binary(BB) tree, (a,b) tree, *Tries- Succint Trie, *Radix Tree, *Ternary Search tree and *Judy arrays.

Note: Since I already have the implementations of many of these data structures and would just need to move to C++ 11, I believe it shouldn't make much time to implement them too and thus, I planned my **tentative timeline** accordingly as **follows**:

01/04/2014 – 06/04/2014 – Looking at the source code of OGDF and understanding its architecture in detail.

07/04/2014 – 13/04/2014 – Learning new data structures marked with asterisk (*) in Dictionary section above.

Phase – I: Implementing Priority Queue using different data structures -

14/04/2014 – 20/04/2014 – Binary heaps.

21/04/2014 – 27/04/2014 – Fibonacci heaps.

28/04/2014 – 04/05/2014 – Mid semester exams II in university. Binomial heaps.

05/05/2014 – 11/05/2014 – Pairing heaps.

12/05/2014 – 18/05/2014 – Heap-On-Top queues.

Phase II: Dictionaries

19/05/2014 – 25/05/2014 – Balanced Search Trees – AVL Trees and Hashing.

26/05/2014 – 29/06/2014 - Skip list – Sorted Sequence

30/06/2014 – 08/06/2014 – End semester examination. Bug fixing or pending work if any.

09/06/2014 – 15/06/2014 – Balanced Binary Trees.

16/06/2014 – 22/06/2014 – Red Black Trees.

Phase III: Review

23/06/2014 – 30/06/2014 – Review of all the data structures implemented above.

I will be following the software development life cycle (SDLC) to complete this project in the given time frame and **testing** and **documentation** for each module or sub-project will be done at the end of their respective week.

01/07/2014 – 10/07/2014 – Implementing Dictionary using Tries and Radix Trees.

11/07/2014-16/07/2014 - **Final Submission.**

Anticipated challenges: Implementing some of the complex data structures marked with asterisk (*) in Dictionary plus implementing them as efficiently as possible.

Programming Experience

Programming Languages: Expert – C++, Java, Good – Ruby, Python.

Operating Systems : primarily Linux, Windows.

Libraries: Used Boost C++

Open source software experience: Contributed by submitting quite a lot of patches and did some enhancements in plasma media center – KDE last year and involved with rack space community.

Academic Experience

Academic Performance: My majors are in computer science and I have undertaken courses like – advanced data structures and algorithms (ADSA) , analysis and design of algorithms (ADA), communication and distributed processes, Operating system, machine learning, etc. I am also the head of Programming Club and Chair, ACM Student Chapter at my university for the past 1.5 years where I have organized a lot of programming competitions at the national level, conducted workshops to give tutorials to freshers and currently developing the Online judge with problem recommendation system at my university for better process of learning.

Do you have any experience with graph algorithm or graph drawing (theory or applications)? And Please describe your knowledge in algorithms and data structures

To best describe my experience with graph algorithms and some advanced algorithms like segment tree with lazy propagation, suffix tree, fenwick tree, etc., I would like to highlight my performance in the ACM – ICPC (International Collegiate Programming Contest), the world's largest and most prestigious programming contest, where I have been the 2 time Asia Regional finalists in years – 2012 and 2013. I have implemented most of the graph algorithms like SCC, Dijkstra's algorithm using min priority queue, MST like Kruskal's ,etc. and various network flows algorithms. The data structures implemented are priority queues, AVL and Red- black trees, binary index tree (BIT), fenwick tree, Suffix Tree, Fibonacci heap, Hashes, etc. To name a few.

Why OGDF?

Have you already used OGDF before GsoC?

No, I haven't used it before. But interested to use it now as this is a great tool as well as a source of learning for professionals and beginners.

Why are you interested in graph algorithms / graph drawing / your particular project?

Though I have implemented many of these algorithms (mainly during the programming contest) many times before, I never used them in most of my projects. I consider it to be a great opportunity to build new tools using these data structures and algorithms. Further, I believe my previous experience in programming will be very helpful in this project and will further hone my

skills in the understanding of various algorithms. I have always been looking for a challenging project in this category and this project of your organization distinguished it from others.

After GSoC, you envision your involvement with OGDF will be:

I will be continuously involved with OGDF in future to do bug fixes, implement new features, etc.

Additional Notes

To get a brief insight of my technical knowledge and experience in programming, I would request you to kindly visit my profiles on various online judges like SPOJ, Codechef, Codeforces, etc. as mentioned on <http://saurabhj.net> and have solved more than 500 challenging problems based on various algorithms, mostly in C++ and Java, on them. I have also done several industrial and academic projects in the past and latest one is the project I did in internship at one of the startups where I developed the full stack in Ruby on Rails. The project I did there was most challenging as I had no experience in rails before plus there were a lot of accomplishments to be done in the given time frame. I am glad to inform that I successfully managed and completed this project and the product has gone live too.