

Open Source + Libraries

Algorithms - OSS Project

Project in Python and Java that has solutions to a lot of common and complex Algorithmic problems (75+ and counting), found on popular Competitive websites like hackerrank.com. Under **BSD License**

Special - Won a Hacktoberfest T-Shirt for this OSS contribution

Github Link:- <https://github.com/tejasnikumbh/Algorithms>

HNFComputation - OSS Project

Open Sourced Project in C that implements HNF Matrix Computation using different algorithms. The complete project can be found at the GitHub link. Under **BSD License**

Github Link:- <https://github.com/tejasnikumbh/HNFComputation>

Computation of Hermite Normal Forms of Integer Matrices

Importance of HNF

Many Integer programming algorithms, or algorithms that compute with modules over \mathbb{Z} , have computation of HNF as a core building block. Computation of HNF over integer matrices of large size and value [value meaning the size of the integer entries in the matrix] is therefore a very important problem to solve.

Definition adopted for HNF

There are various definitions for HNF which present the same idea for an HNF in different ways. Different algorithmic implementations choose to represent HNF in different ways will be stated when the corresponding algorithm is laid out. However, if not mentioned specifically, HNF is mostly the column reduced echelon form. For the naïve Gaussian procedure, I have assumed the lower triangular definition as described below.

For a $m \times n$ matrix A , with entries $a[i][j]$

1. $a[i][j] = 0$ if $1 \leq i < j \leq n$ - Lower triangular
2. $a[i][i] \geq 0$ for $1 \leq i \leq \text{rank}(A)$. - Diagonal Positive
3. $0 \leq a[i][j] < a[i][i]$ for $1 \leq j < i \leq \text{rank}(A)$ - Row entries non negative and less than corresponding diagonal.

RandomWriter - OSS Project

Open Sourced Project in C++ that generates stories. Generative Algorithm using Markov Models that trains on user data of different novella. Using Probabilistic Models, this generates stories that resemble the style of writing of the author. Grammatically correct stories for higher orders of the Markov Model used. Under **BSD License**

Github Link:- <https://github.com/tejasnikumbh/RandomWriter>

ROBDD - OSS Project

Reduced Ordered Binary Decision Diagram. A very important data structure in Computer Science Research for Streaming BDDs & memory management. Implemented in C++ and under **BSD License**.

Github Link:- <https://github.com/tejasnikumbh/ROBDD/tree/master>

Open Source + Libraries

LogicSimulator - OSS Project

Open Sourced Project in C++ that implements logic simulation useful for Circuit Simulation. Useful for simulation of circuits using computers. Open sourced under **BSD License**

Logic-Simulator

Event Driven Logic Simulator for Foundation of Project Course. This simulator is written in c++ and makes use of the naive Twin Queue (Event Processor and Gate Processor) Algorithm to simulate the evaluation of a circuit with logic gates.


Currently the circuit is hardcoded into the code. The contents of this repository are described below.

- Sample Circuit Image of circuit which is simulated in the simulator.
- main.cpp Main program that simulates the circuit.
- graph.h Graph Abstraction that is leveraged for modelling of circuit as a graph. Note that this has minor modifications to the definition of a node and an arc to accomodate the concepts of gate and wires respectively.
- circuitInitializer.h and circuitInitializer.cpp Interface and implementation of circuit initializer code. The code is sufficiently modularised such that the definition of the ckt can be changed easily. Further revisions would include taking in ckt input from a text file.
- circuitSimulator.h and circuitSimulator.cpp Interface and implementation for the circuit simulator. Here is where the core recursive algorithm lies. The algorithm consists of two functions, processEvents and processGates, which recursively call each other until the queue size becomes 0, indicating that there is no need to simulate the ckt further as there is no change in inputs to the next layer.

Github Link:- <https://github.com/tejasnikumbh/LogicSimulator>

Bleeding edge libraries I'm exploring -

README.md



Linux CPU	Linux GPU	Mac OS CPU	Windows CPU	Android
build passing	build passing	build passing	build running	build passing

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets you deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code. TensorFlow also includes TensorBoard, a data visualization toolkit.

TensorFlow - <https://github.com/tejasnikumbh/tensorflow>

State of the art Machine Learning library by Google useful for Neural Networks and other state of the art algorithms involving regression, classification