**Code Walkthrough:**

1. **Main.cpp:**

This file is the starting point for the program. This has the main() function. The command line arguments passed are read here and then are used to create the barnesTree object.   
From here we call the function barnesTree::run(). The time taken to complete the execution of barnesTree::run() is calculated.

1. **barnesTree.cpp:**

This is the file having the implementation for the Barnes-Hut algorithm. The engine of the algorithm is the function **barnesTree::run ()** which is called from the main(). In this function we will first initialise the positions of the particles and store it in the data structure **particleList.** This is done by calling the method **barnesTree::prepareInitList ().** The snapshot of the initial particle positions is created by **barnesTree::logParticlePosition** () function call. It created the snapshot in the folder **output**/  
After the initial positions of the particles are prepared, we will build the Quadtree for the current system. This is done by calling the function **barnesTree::insert ()** for all the particles. When all the particles are inserted we have the Quadtree with **m\_root** being the root of the tree.

Then we compute the center of mass of the particle system. This is done by calling the function **barnesTree::parallelCalcCenterOfMass ()** in the parallel version of the code and by calling **barnesTree::calcCenterOfMass ()** in the serial version of the code.

Till here we are done with the initial setup for the algorithm. Now we will compute the forces and the particles position for the number of timesteps specified by user.

**barnesTree::calcForce ()** computes the total force on the particle. After the force computation, the particles current parameters like position and the velocity are updated by calling the **particle::updateParticle ()** for each particle.

After the completion of a single timestep we will rebuild the tree using the updated particles. This is done by calling the function **barnesTree::rebuildTree ().** Then the center of mass for the new tree is calculated.

After we have run the algorithm for the last timestep we will output the final status of particles to the file in **output/** folder

1. **node.cpp**

The file node.cpp contains all the data members and functions specific to each node.

1. **particle.cpp**

This file contains all the data members and functions specific to the particle.

1. **common.hpp**

The default configurations for the program are stored in this file.