

# Function Descriptions

*Parismita Das*

*24 February 2017*

## General Description for HeapHop Implementation

The HeapHop function takes in the Distance Matrix diagonal elements along with its width and no of rows as input. This input is converted to matrix in PseudoMatrixRcpp.cpp file.

The PseudoMatrix Class object has the Cluster Classes information, Index and Data as attributes. FusionnedClass functions calculate informations such as fusioncost, available index, No of clusters and ColArray calculates the value in Data. GeneralFunctions class object prints the Matrix. ClassesHeap is where actual Heap formation and and Output Heap calculation is done, which is then merged in the file PseudoMatrixRcpp.cpp for final Output.

## Function Description

### 1) GeneralFunctions.cpp

- MyPrint : It prints the given Matrix

### 2) ColsArray.cpp

- Restructure : For PseudoMatrix object, It stores the column index and data content of the Matrix.
- CheckMe : Checks if the Matrix element exist on a particular memory
- Initialize : Initializes matrix M and points next memory address of M via ColIndex and Data
- Set : There are two set functions using concept of function overloading. In `Set(int Col, double Value, int ForceIndex, bool Assumption)` The Col and Value are stored in the forced index and `Set(int Col, double Value)` is called. In `Set(int Col, double Value)` New memory is allocated to the given input column and value is set to data.
- Value : In `Value(int Col, int SupposedPlace)` The column index is checked if it is same as SupposedPlace index and Data is returned. In `Value(int Col)` The Data in the Col index is returned.

### 3) FusionedClasses.cpp

- Initialize : initialises current index, previous index and next index
- InitializeFusionCost : initialises Fusion Cost depending on the Adjacent index's Value
- MyValue : return value for current index
- Swallow : Deletes current index, sets next index as current, increase the number of fusionned classes by 1. It basically means one cluster is being made.
- ComputeMyFusionCost : calculate fusion cost depending on the Adjacent index's Value using the Lance and William Algorithm.
- Exist : checks if My Index is euqal to Available Index, and return bool

- MyCardinal : return Number of fusion classes : `NbFusions + 1`, ie the parameter for ward's criterion.

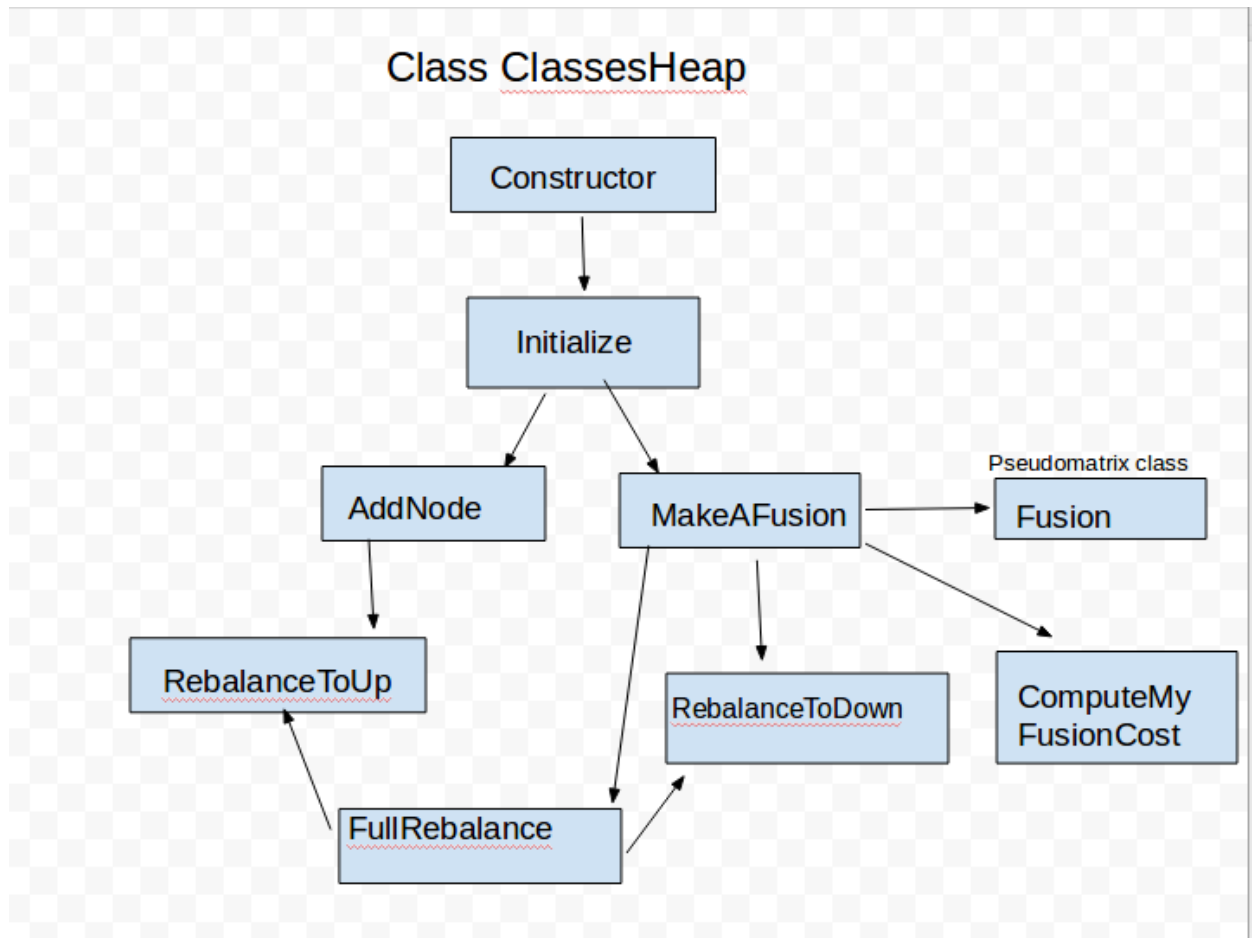
#### 4) PseudoMatrix.cpp

- Initialize : Takes pseudomatrix object as attribute and initialises cluster classes, FusionCost and the Matrix Data.
- Set : sets data according to index given, calls Set function of colsArray.cpp
- Value : sets index according to index given, calls Value function of colsArray.cpp
- Fusion : Takes next index, merges the index according to its cost.
- DisplayMatrixA : display matrix

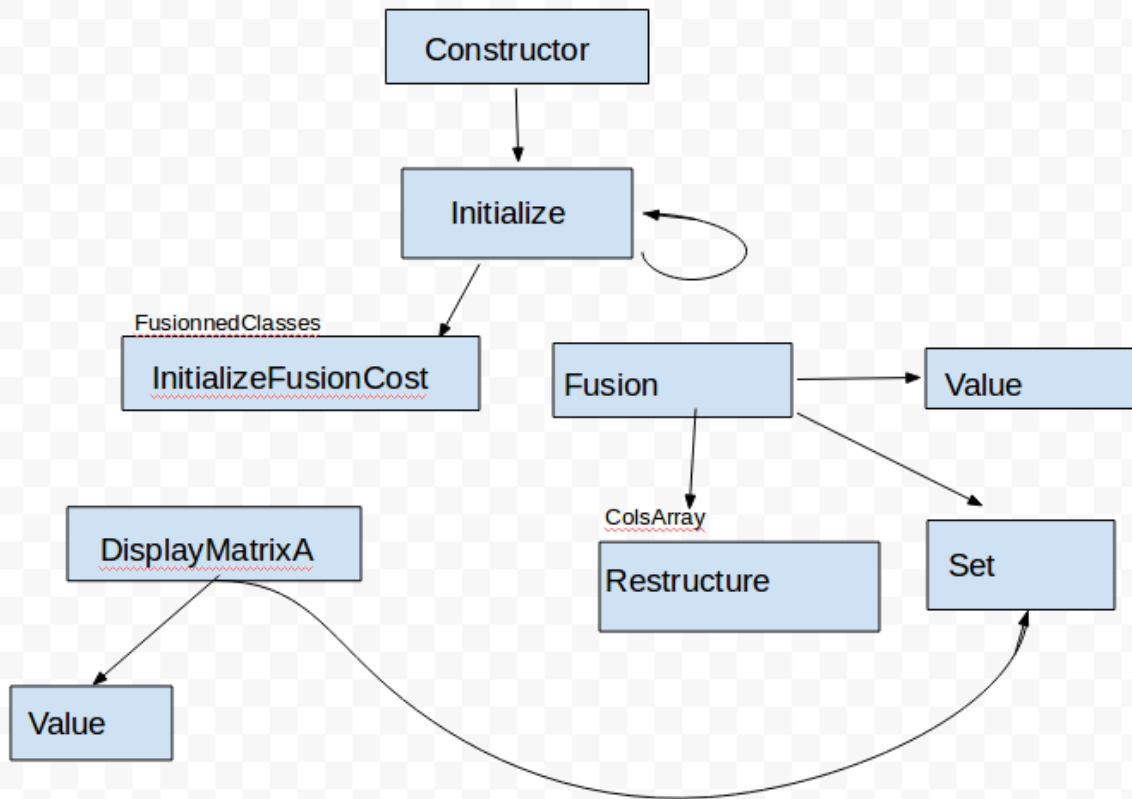
#### 5) ClassesHeap.cpp

- CheckMe : checks validity of left right element of heap
- Initialize : initialises the heap, output, make clusters of given number,
- Swap : swap two elements
- RebalanceToDown : checks the lower right to larger and swap the elements to maintain min heap property
- RebalanceToUp : checks upper element to be smaller and swap to maintain heap property
- AddNode : adds node to heap and rebalances
- MakeAFusion : make clusters of given number by merging current clusters
- FullRebalance : rebalances to down at the beggining and when rebalance to up not possible

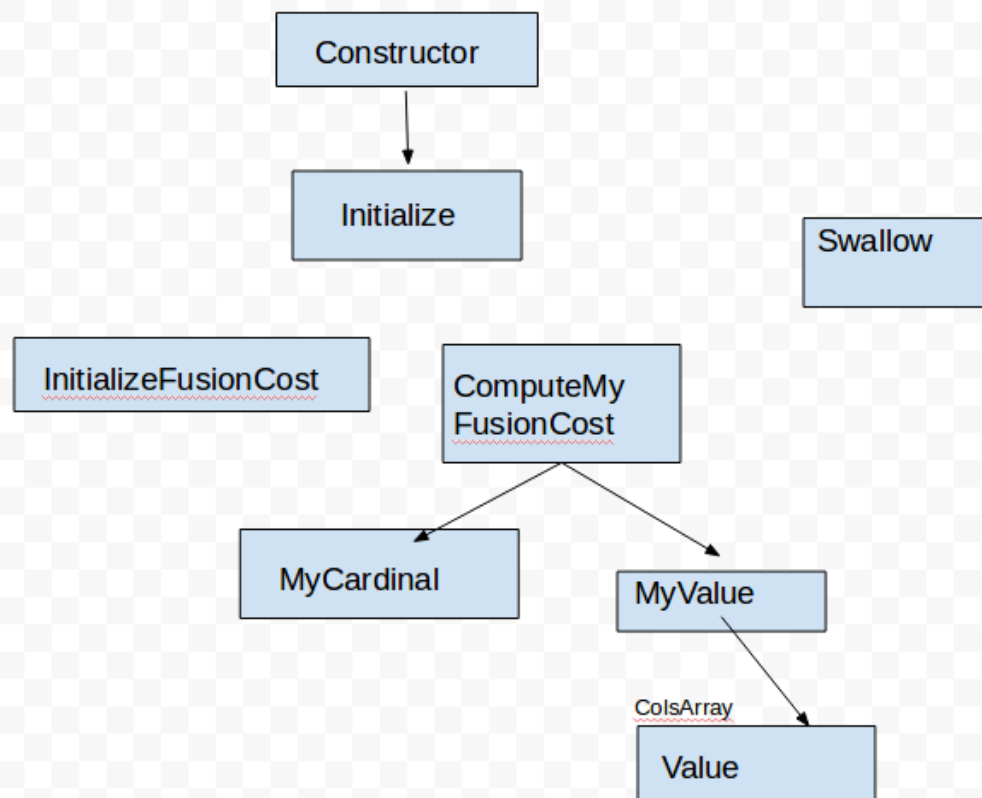
## The Functions Dependency Graph



## Class PseudoMatrix



## Class FusionedMatrix



## Class ColsArray

