**REPORT**

Name : U. Krishna chaitanya

UFID : 3855-8149

Email : uppalak@ufl.edu

**Global variables :**

struct node{   // creating a struct to hold the values and degrees

   int data,degree;

   node \*next,\*prev,\*child,\*parent;

   bool cutchild;

};

node \*head;

node \*maxpointer;   // max pointer to the fibonacci heap

**Functions and Pseudocode :**

|  |  |  |
| --- | --- | --- |
| **node \*create(int myValue)** | | |
| Description | Creates node with all fields | |
| Parameters | int | Data in the node |
| Return Value | No return value | |

|  |  |  |
| --- | --- | --- |
| **void inserttolist(node \*store)** | | |
| Description | Inserts records into the fibonacci heap. | |
| Parameters | temp | Pointer to  create node |
| Return Value | No return value | |

|  |  |  |
| --- | --- | --- |
| **void meld(node\* first,node\* second)** | | |
| Description | Combines two records such that one becomes child of another. | |
| Parameters | first | 1st node |
| second | 2nd node |
| Return Value | No return value | |

|  |  |  |
| --- | --- | --- |
| **void cascadingcut(node\* parent)** | | |
| Description | remove the record given and insert as root | |
| Parameters | parent | Initially called with parent of node that has been moved due to increasing key |
| Return Value | No return value | |

|  |  |  |
| --- | --- | --- |
| **void increasekey(node \*temp,int value)** | | |
| Description | increase value, removes node if it is greater than parent and calls cascadingcut() if necessary | |
| Parameters | temp | Pointer to the node to be incremented (you get that from hashmap) |
| value | How much value you need to increase |
| Return Value | No return value | |

|  |  |
| --- | --- |
| **void removemax()** | |
| Description | Remove the max element in the tree and point the max to next maximum |
| Return Value | No return value |

|  |  |  |
| --- | --- | --- |
| **int main(int argc, char\*\* argv)** | | |
| Description | Main function to handle input file and operations to be called according to input file and given a output | |
| Parameters | argc | Number of input parameters |
| argv | Each paramerter |
| Return Value | 0 if it is successfully finishes | |