Report of Project COP5536 Advanced Data Structure

Daoyu Tu 91473473

1. Programming Environment

Visual Studio 2013

2. Function Prototypes

In AdjGraph.h:

AdjGraph(int n=0)
//Create an array of adjacency lists.
void addEdge(int src, int dest, int w)
//Add an edge from src to dest. A new node is added to the adjacency list of src. The node is added at the beginning.

In BinaryTrie.h & BinaryTrie.cpp:

bool insert(string str, int np)
//Insert a node into the binary trie, the str is the IP and the np is the nextHop, return false if failed to insert.
trieNode *search(string str)
//Search with IP address as a string, return the node that matches the IP address.
bool Merge()
//Merge the nodes that have the same nextHop and have the same parent.

In Dijkstra.h:

void ShortestPaths(AdjGraph *graph, int v, int *prev,int *dist,int *nextHop)
//Graph and source node v are given, then *prev, *dist, *nextHop are calculated. In array prev, prev[i] stores the previous node before i in the path from v to i. In array dist, dist[i] stores the distance of the path from v to i. In array nextHop, nextHop[i] stores the next node after v in the path from v to i.

In Fibonacci.h:

heapNode<V>* insert(V value)
//Insert node value into the fibonacci tree.
void decreaseKey(heapNode<V>* n,V value)
//Set node n to value.
V removeMinimum()
//Remove the min value node from the tree and return this value.
void merge(FibonacciHeap& other)
//Merge this heap with other heap.

In ssp.cpp:

int main(int argc, char *argv[])
//Takes 3 parameters, graph data file path, starting node and ending node. Apply Dijkstra algorithm on Fibonacci tree to calculate shortest path, print the cost of the path and all the node in the path.

In routing.cpp:

int main(int argc, char *argv[])
//Takes 4 parameters, graph data file path, IP address data file path, starting node and ending node. Use Binary Trie to store the IP address and apply Dijkstra algorithm to calculate shortest path print the cost of the path and all IP address in the path.

3. How to Compile and Run

(1) unzip the code
(2) cd to the root of the project
(3) make ssp
(4) make routing
(5) cd bin

(6) ./ssp parameter1 parameter2 parameter3 to run the first part(parameter1 is the graph input file name, parameter2 is the starting node index, parameter3 is the ending node index)
(7) ./routing parameter1 parameter2 parameter3 parameter4 to run the second part(parameter1 is the graph input file name, parameter2 is the IP address input file name, parameter3 is the starting node index, parameter4 is the ending node index)

4. Structure of the Program

./src : contatins source files
./bin : contains object files
./data : contains the test data files
./Makefile : makefile to build the project