Project Report - ADS

Name: SHUBHI JAIN
UFID: 1159-7408
E-mail: shubhijain106@ufl.edu

Instructions:
Compiler: Xcode IDE

To compile:
$make

To run part 1,command line:
$ssp file_name source_node destination_node

To run part 2,command line:
$routing file_name_1 file_name_2 source_node destination_node

Function prototypes:-

1.Main(): *Main* takes input from the command line and stores in corresponding variables. Each router has an IP address and packets are forwarded to the next hop router by longest prefix matching using a binary trie. For each router R in the network, Shortest path Function is called to obtain shortest path from R to each destination router Y. To construct the router table for R, for each destination Y, examine the shortest path from R to Y and determine the router Z just after R on this path.

2.Shortest Path(): It implements Dijkstra's Single Source Shortest Path (ssp) algorithm for undirected graphs using Fibonacci heaps using the adjacency list representation for graphs. This gives a set of pairs <IP address of Y, next-hop router Z>,these pairs are then inserted into a binary trie.

3.Fibonacci Heap(): This class constructs Fibonacci heap with the help of specification of the root.
Functions in Fibonacci class are:

3a.)check_empty():checks if initially the heap is empty.

3b.)add vertex():i inserting nodes in the heap with key values. Elements are inserted in circular list that stores all the roots in the heap.

3c.)decrease key(): It is to decrease the key of a node by a given amount and check if the minimum pointer should point to this node or not.

3d.)find minimum(): It is to find minimum element in the heap which is simply found by returning the minimum pointer.

3e.)delete minimum():removes the element pointed by the MINpointer i.e. the node with the minimum key. Left and right sibling nodes' pointers are rearranged to maintain the circular list after the removal. If the removed node had any children, they are added in the root list. After that, roots with same degree are combined together and new min node is calculated

3f.)Pairwise combine():a degree table is kept which helps to carry out the pairwise combine process.

4.Node(): *This* class defines a node with fields such as child,parent,left sibling,right sibling,degree and key.

Functions in Node class are:

4a.)Add child():This function adds a child to a node if it does not have any child before,otherwise calls the sibling method and increases the degree of the node.

4b.)Add sibling(): This function adds a sibling to a node if it has children.It inserts a new node in rightmost end.

4c.)del():if we want to arbitrary remove a node then its parent ,left and rightmost sibling pointers and then sibling nodes will change accordingly.

4d.)lsible and risible():used to fetch rightmost and leftmost siblings of a node.

4e.)insert_oedge() and insert_iedge():used to insert incoming and outgoing edges of a node in vectors.

5.Edges(): This class defines an edge between a start and end node and also specifies the length of that edge.

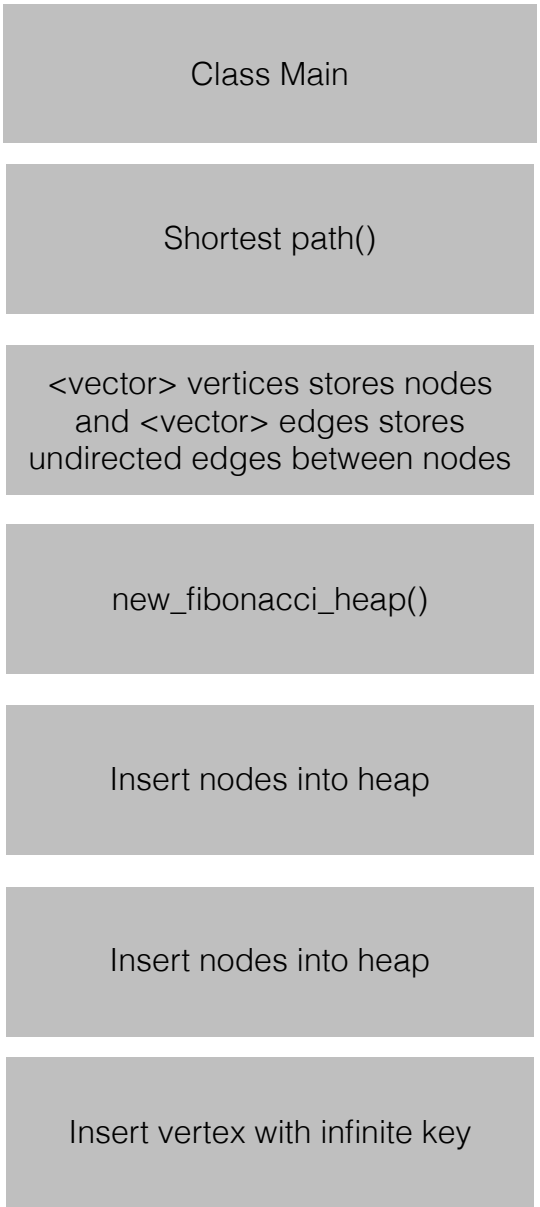
6.Tries(): Takes ip address and build a trie.Structure of trie consists of child pointers and parent pointer.

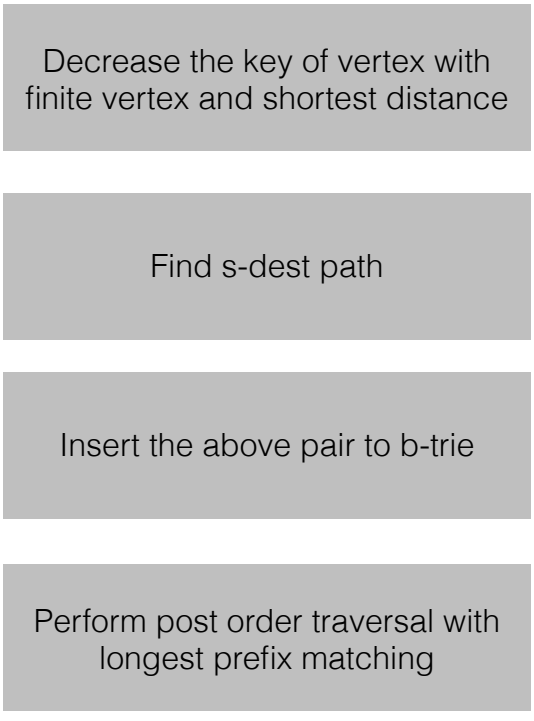
6a.)Traversal(): Finally, a postorder traversal, removes subtries in which the next hop is the same for all destinations. Thus, multiple destinations having a prefix match and the same next hop are grouped together in the trie.

6b.)Insert():inserts shortest path in tries.

Structure:

Fibonacci heap :: O(nlogn+e)





Part 1:Compilation and output:

```
SHUBHIs-MacBook-Air:~ Shubhi$ cd Pictures
SHUBHIs-MacBook-Air:Pictures Shubhi$ g++ ssp.cpp -o ssp
ssp.cpp:133:8: warning: 'this' pointer cannot be null in well-defined C++ code;
      comparison may be assumed to always evaluate to false
      [-Wtautological-undefined-compare]
    if(this == NULL)
       ^~~~     ~~~~
ssp.cpp:143:8: warning: 'this' pointer cannot be null in well-defined C++ code;
      comparison may be assumed to always evaluate to false
      [-Wtautological-undefined-compare]
    if(this == NULL)
       ^~~~     ~~~~
2 warnings generated.
SHUBHIs-MacBook-Air:Pictures Shubhi$ ./ssp input1.txt 0 4999
214
0 4822 1891 2767 1942 4964 1927 4999
SHUBHIs-MacBook-Air:Pictures Shubhi$
```

Part 2: Compilation and output

```
Last login: Wed Apr 15 10:32:38 on ttys000
SHUBHIs-MacBook-Air:~ Shubhi$ cd Pictures
SHUBHIs-MacBook-Air:Pictures Shubhi$ g++ routing.cpp -o routing
routing.cpp:180:8: warning: 'this' pointer cannot be null in well-defined C++
      code; comparison may be assumed to always evaluate to false
      [-Wtautological-undefined-compare]
    if(this == NULL)
       ^~~~    ~~~~
routing.cpp:191:8: warning: 'this' pointer cannot be null in well-defined C++
      code; comparison may be assumed to always evaluate to false
      [-Wtautological-undefined-compare]
    if(this == NULL)
       ^~~~    ~~~~
2 warnings generated.
SHUBHIs-MacBook-Air:Pictures Shubhi$ ./routing graph.txt ip.txt 0 3
3
11000000000000101010100000001 11000000000000101010100000000010 11000000000000101
010100000000100 SHUBHIs-MacBook-Air:Pictures Shubhi$
```