First, I write a structure of Node. Every Node has a character object ‘chara’(excluding root), a vector ‘N’ which contains pointers that pointer to other Node. A boolean object ‘isWord’ to show whether current word is a word. A vector ‘netAdd’ contains website address if current word is a word.

In the class Tries, I wrote two functions. There are insertWord(string str, int netAddress) and queryWords (string str).

In the insertWord(string str, int netAddress) function, first, I compared the first character of the input string with the ‘chara’ of the Node which is in the root’s N vector. If it can find this character, now this Node become root, I compared the second character of the input string with the ‘chara’ of the Node which is in the current root’s N vector, and so on. Once, it can’t find the current character in current root’s N vector, I allocate a new Node whose ‘chara’ is current character of the input string, then push it in the current root’s N vector. In this process, once, the last character of the input string has been put into a Node. The ‘isWord’ of this Node became ‘true’ and put the inverted index of websit address in the ‘netAdd’ vector of this Node. Finally, finish this function.

After words of all page have been put into this function, we get a tries.

I use an inverted index. I have an array storing the occurrence lists of the website address.

A ‘netAdd’ vector which storing the index of website of the associated term.

In the queryWords(string str) function, I find the first character of the input string with the ‘chara’ of the Node which is in the root’s N vector. If it can find this character, now this Node become root, I compared the second character of the input string with the ‘chara’ of the Node which is in the current root’s N vector, and so on. Once the last character of the input string has been searched and isWord of current Node is true, return the ‘netAdd’ vector.