CS6040 – Router Architectures and Algorithms, Jul.–Nov. 2024
Prof. Krishna Sivalingam
Lab 5 - IP Packet Classification
Due Date: Nov. 9, 2024, 11 PM, BY Moodle
Late submission Policy: 8% per 12-hour duration for a max. of 48 hours

The objective is to implement and understand **Packet Classification Algorithms**.

1 Packet Classification

The objective of this lab exercise is to implement the following packet classification algorithm for IPv4: **Hierarchical Tries** approach (described in Chapter 16 of Medhi/Ramasamy's book) for two-dimensional classification and class notes.

The algorithms can be implemented using C/C++/Java/Python. If existing STLs (templates) are used, sources and location of sources MUST be provided in the report; else, you will earn *NO* credit for this assignment.

Command Line Invocation

```
% ./spt -p RuleFile -i InputAddrFile -o OutputFile
```

1.1 Input Files

The maximum number of bits in each prefix is $W = 16$. The minimum number of bits in the prefix is 8. The maximum number of entries in the RuleFile is 300,000; the maximum number of entries in the InputAddrFile is 1,000,000.

There are two input files:

- RuleFile: Three entries per line; the first entry is the rule number. There are two prefix entries, each of which has two fields: IP address prefix (in IP dotted decimal format; 2 decimal numbers) and prefix length (in bits). For example:

1	128.16.0.0	12	130.205.0.0	16
2	112.36.0.0	14	112.0.0.0	8
	...	...		

- InputAddrFile: Two entries per line; each entry specifies an IP address value (in IP dotted decimal format; 2 decimal numbers). The objective is to determine which rule(s), if any, matches the input pair. For example:

  128.18.1.1 130.205.1.4
  112.38.2.4 112.34.3.3
  ...

1.2 Output Files

There is one output file, with one entry per line as follows:

Address 1	Address 2	No. of Matches	Rules matched	Search Time (μs)
128.18.1.	130.205.1.4	1	1	45.6
112.38.2.4	112.34.3.3	1	2	25.4

At the end of the file, there will be one line printing:

```
Average Search Time is: ... microseconds
```

1.3 What to Submit

- All Source files
- One set of input/output files with RuleFile having 1000 rules and input address file with at least 100 addresses.
- Sample Input Files will be provided; you are also expected to generate your own set of input files using the generator code provided.

1.4 Grading

- Algorithm implementation and correct execution on test files: 90 marks
- Viva Voce Exam: 10 marks

No README: -5%; No Makefile: -5%; No Sample Output Files: -10%