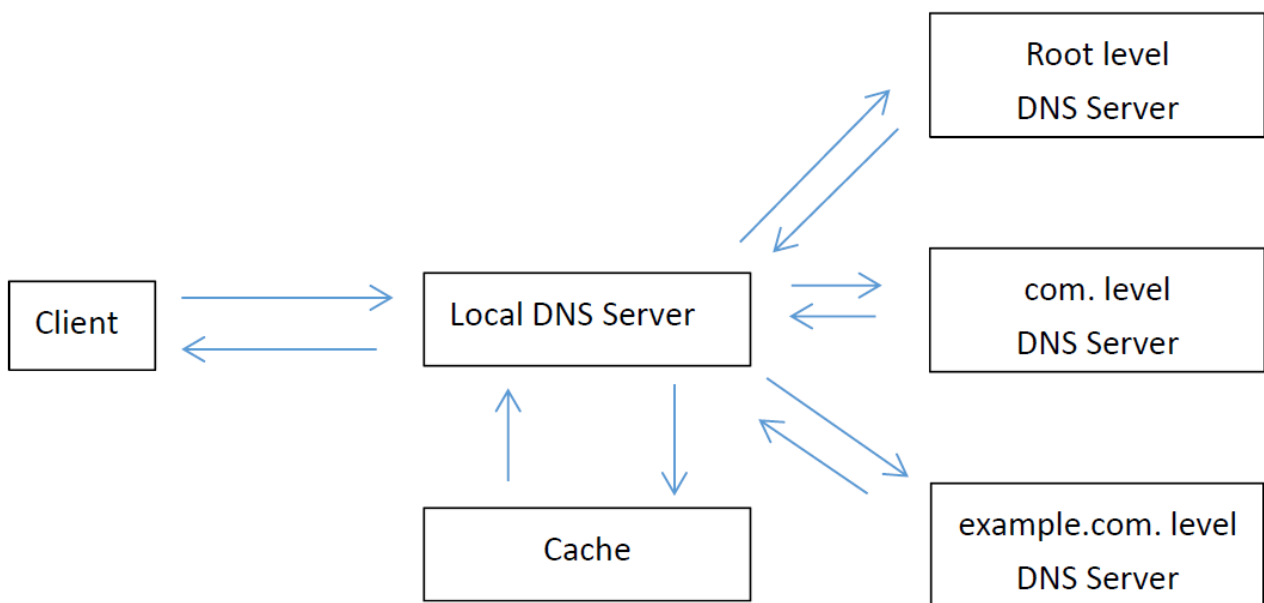


CS305 2022 Spring Assignment 1

A **local DNS server** is a server which is setup inside of a company, homes network, or residential ISP for address/name translations. The goal of this assignment is to implement a **local DNS server** in Python.

IMPORTANT NOTES:

- Your python code should be named as "LocalDNSServer.py" using UTF-8 as file encoding.
- The local DNS server should **listen on port 5533** (do NOT change it!!!) and **IP address 127.0.0.1** (do NOT change it!!!) .
- When querying, the RD in DNS message should be set to 0* (or set the flag to be 0x0000), which means that your DNS query is **iterative**.
- **Comments in code is MUST.**
- A basic template will be provided and you need to implement the functionalities in the **TODO** fields. You may choose to not use the template, but you need to confirm your solutions match the corresponding input and output as shown in the examples below.
- We will use **dig** as the client in this assignment.



Environments

- Python 3.9.7
- dnspython and dnslib libraries (optional, but recommended and used in the template)
- You may use other libraries, but you must ensure that your query procedure is **iterative**

Implementation detail

1. The local DNS server should listen on **port 5533** and **IP 127.0.0.1**. The main functionalities of the local DNS server include:

- (1) Listen and accept DNS queries from client, support default query type: A type.
 - (2) Maintain a Cache: ① Cache should be human-readable, following the RR format: (name, value, type, ttl).
 - ② A new query should be recorded if not in the cache already.
 - (3) Maintain a list of records about the domain names and IPv4 addresses of Root DNS servers.
 - (4) Search related DNS responses in cache according to the received query.
 - (5) Invoke DNS queries to other DNS servers if needed and cache the new response.
 - (6) Reply to client with the response.
2. Take a DNS query to "www.example.com" as an example what your need to implement is about:
- (1) The client sends a DNS query about "www.example.com" to the local DNS server.
 - (2) The local DNS server searches its mapping in the cache.
 - if the mapping is found in Cache, the local DNS server replies to the client as a DNS response object.
 - If a mapping is not found, then a translation is resolved **iteratively** as follows:
 - (a) the local DNS server sends the query to **root level DNS Server**, and gets a DNS response message about the list of .com level DNS servers.
 - (b) the local DNS server then sends a query to **.com level DNS server**, and gets the DNS response message about the list of example.com level name servers.
 - (c) the local DNS server then sends the query to **example.com. level DNS Server**, and gets the DNS response message about the mapping of the original DNS query.
 - (d) the local DNS server caches the mapping in cache.
 - (e) the local DNS server replies to the client with the mapping.
3. Grading criteria
- Basic iterative query to get a name/address translation (80)
 - Caching (20)
 - Multi-threading local DNS server and concurrency control (bonus, up to 20)
 - Implement a multi-thread version local DNS server that can serve multiple queries concurrently.
 - Provide at least one implementations to tackle the problem when several clients access the server in the same time, which means that you should ensure the thread safety in cache, since it could be overwritten by multi threads at the same time.
5. NOTE again: in the DNS message's flag, RD should be set to 0 which means it does not query recursively!

Test detail

Input format

There are two places that you need to input.

- The first one is when you run the LocalDNSServer.py , you need to input the ip address of your network interface card.

- The second one is when you open dig, you should input a dig DNS request, e.g.,
dig @"your ipv4 address" "queried domain name" a -p "your port"
Here please specify the port as 5533

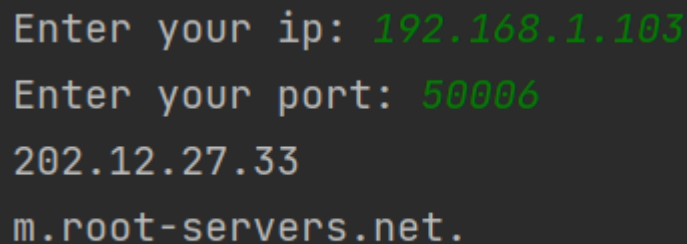
Output format

- The output format is standard DNS message format, you can use `dnslib library` to construct and resolve it (we have this example in the template).
- When the DNS mapping is read from cache you should print "read from cache" on the console ("" is not need)

Test process

- ① Run LocalDNSServer.py.
- ② Enter IP address (you can use ipconfig to search your IP address) and port (this port is arbitrary but you should avoid conflict with other process) in console. If success, it will print the information about the root DNS server.

This means that you can use this IP address and port to complete and resolve a basic DNS query.



```
Enter your ip: 192.168.1.103
Enter your port: 50006
202.12.27.33
m.root-servers.net.
```

- ③ Open another command line and input the command:

dig @127.0.0.1 www.baidu.com a -p 5533

Then the console should print your local DNS server's response.

```
dig @127.0.0.1 www.baidu.com a -p 5533

; <<>> DiG 9.16.26 <<>> @127.0.0.1 www.baidu.com a -p 5533
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55638
;; flags: qr rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.baidu.com.                IN      A

;; ANSWER SECTION:
www.baidu.com.                 60      IN      CNAME   www.a.shifen.com.
www.a.shifen.com.             60      IN      A       14.215.177.38

;; Query time: 450 msec
;; SERVER: 127.0.0.1#5533(127.0.0.1)
;; WHEN: Sat Mar 05 08:10:49 ;; MSG SIZE rcvd: 74
```

and if it is read from cache, you should print "read from cache" in the python console where your server runs

If you resolve a domain name and get a result in cname type, you should keep that result and include it in the response

The above output example shows

- when you search www.baidu.com and find a cname type record of www.a.shifen.com,
- you need to print www.a.shifen.com and then use www.a.shifen.com to query the final response, which is shown as the above screenshot.

④ We will test about more than 4 testcases continuously.

What to submit

- You should provide your implementation in the file LocalDNSServer.py and another file such as the cache if needed.
- Please provide detailed comments in your code to explain different parts of your code. You need to at least illustrate in your implementation about basic part, cache part, and concurrency part (if any).
- A PDF file that includes some necessary screenshots to show your code works and some brief explanations.

TIPS

The below WireShark screenshots show the process about resolving a DNS query **iteratively**.

① query 8.8.8.8 and get records of root DNS server in NS type

20	7.173352	192.168.1.104	8.8.8.8	DNS	59 Standard query 0xe4fd NS <Root>
21	7.219875	8.8.8.8	192.168.1.104	DNS	270 Standard query response 0xe4fd NS <Root> NS a.root-ser

query:

Domain Name System (query)					
Transaction ID: 0xe4fd					
> Flags: 0x0000 Standard query					
Questions: 1					
Answer RRs: 0					
Authority RRs: 0					
Additional RRs: 0					
Queries					
> <Root>: type NS, class IN					
[Response In: 21]					

answer:

Domain Name System (response)					
Transaction ID: 0xe4fd					
> Flags: 0x0000 Standard query response, No error					
Questions: 1					
Answer RRs: 13					
Authority RRs: 0					
Additional RRs: 0					
Queries					
> <Root>: type NS, class IN					
Answers					
> <Root>: type NS, class IN, ns a.root-servers.net					
> <Root>: type NS, class IN, ns b.root-servers.net					
> <Root>: type NS, class IN, ns c.root-servers.net					
> <Root>: type NS, class IN, ns d.root-servers.net					
> <Root>: type NS, class IN, ns e.root-servers.net					
> <Root>: type NS, class IN, ns f.root-servers.net					
> <Root>: type NS, class IN, ns g.root-servers.net					
> <Root>: type NS, class IN, ns h.root-servers.net					
> <Root>: type NS, class IN, ns i.root-servers.net					
> <Root>: type NS, class IN, ns j.root-servers.net					
> <Root>: type NS, class IN, ns k.root-servers.net					
> <Root>: type NS, class IN, ns l.root-servers.net					
> <Root>: type NS, class IN, ns m.root-servers.net					
[Request In: 20]					
[Time: 0.046523000 seconds]					

② query 8.8.8.8 and get IP address of root DNS server

22	7.234943	192.168.1.104	8.8.8.8	DNS	78 Standard query 0xf950 A a.root-servers.net
23	7.285914	8.8.8.8	192.168.1.104	DNS	94 Standard query response 0xf950 A a.root-servers.net A

query:

Domain Name System (query)					
Transaction ID: 0xf950					
> Flags: 0x0000 Standard query					
Questions: 1					
Answer RRs: 0					
Authority RRs: 0					
Additional RRs: 0					
Queries					
> a.root-servers.net: type A, class IN					
[Response In: 23]					

answer:

Domain Name System (response)					
Transaction ID: 0xf950					
> Flags: 0x0000 Standard query response, No error					
Questions: 1					
Answer RRs: 1					
Authority RRs: 0					
Additional RRs: 0					
Queries					
> a.root-servers.net: type A, class IN					
Answers					
> a.root-servers.net: type A, class IN, addr 198.41.0.4					
[Request In: 22]					
[Time: 0.050971000 seconds]					

③ query root DNS server and get IP address of top-level DNS server

24	7.291027	192.168.1.104	198.41.0.4	DNS	73 Standard query 0x0d0a A www.baidu.com
25	7.516425	198.41.0.4	192.168.1.104	DNS	533 Standard query response 0x0d0a A www.baidu.com NS e.gtld-servers.net NS b.gtld-servers.net N

query:

```

Domain Name System (query)
Transaction ID: 0x0d0a
> Flags: 0x0000 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
> www.baidu.com: type A, class IN
[Response In: 25]

```

answer:

```

Domain Name System (response)
Transaction ID: 0x0d0a
> Flags: 0x8200 Standard query response, No error
Questions: 1
Answer RRs: 0
Authority RRs: 13
Additional RRs: 11
Queries
> www.baidu.com: type A, class IN
Authoritative nameservers
> com: type NS, class IN, ns e.gtld-servers.net
> com: type NS, class IN, ns b.gtld-servers.net
> com: type NS, class IN, ns j.gtld-servers.net
> com: type NS, class IN, ns m.gtld-servers.net
> com: type NS, class IN, ns i.gtld-servers.net
> com: type NS, class IN, ns f.gtld-servers.net
> com: type NS, class IN, ns a.gtld-servers.net
> com: type NS, class IN, ns g.gtld-servers.net
> com: type NS, class IN, ns h.gtld-servers.net
> com: type NS, class IN, ns l.gtld-servers.net
> com: type NS, class IN, ns k.gtld-servers.net
> com: type NS, class IN, ns c.gtld-servers.net
> com: type NS, class IN, ns d.gtld-servers.net
Additional records
> e.gtld-servers.net: type A, class IN, addr 192.12.94.30
> e.gtld-servers.net: type AAAA, class IN, addr 2001:502:1ca1::30
> b.gtld-servers.net: type A, class IN, addr 192.33.14.30
> b.gtld-servers.net: type AAAA, class IN, addr 2001:503:231d::2:30
> j.gtld-servers.net: type A, class IN, addr 192.48.79.30
> j.gtld-servers.net: type AAAA, class IN, addr 2001:502:7094::30
> m.gtld-servers.net: type A, class IN, addr 192.55.83.30
> m.gtld-servers.net: type AAAA, class IN, addr 2001:501:b1f9::30
> i.gtld-servers.net: type A, class IN, addr 192.43.172.30
> i.gtld-servers.net: type AAAA, class IN, addr 2001:503:39c1::30

```

④ query top-level server and try to find authoritative domain name server

33 7.950684	192.168.1.104	192.12.94.30	DNS	73 Standard query 0x6f78 A www.baidu.com
36 8.175712	192.12.94.30	192.168.1.104	DNS	299 Standard query response 0x6f78 A www.baidu.com NS ns2.baidu.com NS ns3.baidu.com NS ns4.baid

query:

```

> Frame 33: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface \Device\NPF_{5A42BE07-848B-4692-BDDB-9B8394F043B9}, id 0
> Ethernet II, Src: Microsof_6d:4c:2b (70:bc:10:6d:4c:2b), Dst: Tp-LinkT_5f:cc:66 (54:75:95:5f:cc:66)
> Internet Protocol Version 4, Src: 192.168.1.104, Dst: 192.12.94.30
> User Datagram Protocol, Src Port: 55555, Dst Port: 53
Domain Name System (query)
Transaction ID: 0x6f78
> Flags: 0x0000 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
> www.baidu.com: type A, class IN
[Response In: 36]

```

answer:

```
Domain Name System (response)
Transaction ID: 0x6f78
> Flags: 0x8000 Standard query response, No error
Questions: 1
Answer RRs: 0
Authority RRs: 5
Additional RRs: 7
Queries
> www.baidu.com: type A, class IN
Authoritative nameservers
> baidu.com: type NS, class IN, ns ns2.baidu.com
> baidu.com: type NS, class IN, ns ns3.baidu.com
> baidu.com: type NS, class IN, ns ns4.baidu.com
> baidu.com: type NS, class IN, ns ns1.baidu.com
> baidu.com: type NS, class IN, ns ns7.baidu.com
Additional records
> ns2.baidu.com: type A, class IN, addr 220.181.33.31
> ns3.baidu.com: type A, class IN, addr 112.80.248.64
> ns4.baidu.com: type A, class IN, addr 14.215.178.80
> ns1.baidu.com: type A, class IN, addr 110.242.68.134
> ns7.baidu.com: type A, class IN, addr 180.76.76.92
> ns7.baidu.com: type AAAA, class IN, addr 240e:940:603:4:0:ff:b01b:589a
> ns7.baidu.com: type AAAA, class IN, addr 240e:bf:b01:1002:0:ff:b024:26de
[Request In: 33]
[Time: 0.275078000 seconds]
```

Hint: in this step you may find many **potential routes** or even not any mappings for further search. One recommendation is that you can save all search paths as a tree and make a depth-first search (DFS) in that tree.

⑤ by iteratively querying, finally get a mapping of A type

37 8.182512	192.168.1.104	220.181.33.31	DNS	73 Standard query 0xe79a A www.baidu.com
38 8.223700	220.181.33.31	192.168.1.104	DNS	326 Standard query response 0xe79a A www.baidu.com CNAME www.a.shifen.com NS ns4.a.shifen.com NS
39 8.232952	192.168.1.104	110.242.68.42	DNS	76 Standard query 0x992f A www.a.shifen.com
40 8.280738	110.242.68.42	192.168.1.104	DNS	334 Standard query response 0x992f A www.a.shifen.com A 14.215.177.38 A 14.215.177.39 NS ns1.a.sl

query:

```
Domain Name System (query)
Transaction ID: 0xe79a
> Flags: 0x0000 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
> www.baidu.com: type A, class IN
[Response In: 38]
```

answer:

```
Domain Name System (response)
Transaction ID: 0xe79a
> Flags: 0x8000 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 5
Additional RRs: 7
Queries
> www.baidu.com: type A, class IN
Answers
> www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
Authoritative nameservers
> a.shifen.com: type NS, class IN, ns ns4.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns5.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns3.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns2.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns1.a.shifen.com
Additional records
> ns1.a.shifen.com: type A, class IN, addr 110.242.68.42
> ns2.a.shifen.com: type A, class IN, addr 220.181.33.32
> ns3.a.shifen.com: type A, class IN, addr 112.80.255.253
> ns4.a.shifen.com: type A, class IN, addr 14.215.177.229
> ns5.a.shifen.com: type A, class IN, addr 180.76.76.95
> ns5.a.shifen.com: type AAAA, class IN, addr 240e:bf:b01:1006:0:ff:b04f:346b
> ns5.a.shifen.com: type AAAA, class IN, addr 240e:940:603:a:0:ff:b08d:239d
[Request In: 37]
```

query:

```
Domain Name System (query)
Transaction ID: 0x992f
> Flags: 0x0000 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
> www.a.shifen.com: type A, class IN
[Response In: 40]
```

answer:

```
Domain Name System (response)
Transaction ID: 0x992f
> Flags: 0x8400 Standard query response, No error
Questions: 1
Answer RRs: 2
Authority RRs: 5
Additional RRs: 7
Queries
> www.a.shifen.com: type A, class IN
Answers
> www.a.shifen.com: type A, class IN, addr 14.215.177.38
> www.a.shifen.com: type A, class IN, addr 14.215.177.39
Authoritative DNS servers
> a.shifen.com: type NS, class IN, ns ns1.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns2.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns3.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns4.a.shifen.com
> a.shifen.com: type NS, class IN, ns ns5.a.shifen.com
Additional records
> ns1.a.shifen.com: type A, class IN, addr 110.242.68.42
> ns2.a.shifen.com: type A, class IN, addr 220.181.33.32
> ns3.a.shifen.com: type A, class IN, addr 112.80.255.253
> ns4.a.shifen.com: type A, class IN, addr 14.215.177.229
> ns5.a.shifen.com: type A, class IN, addr 180.76.76.95
> ns5.a.shifen.com: type AAAA, class IN, addr 240e:bf:b801:1006:0:ff:b04f:346b
> ns5.a.shifen.com: type AAAA, class IN, addr 240e:940:603:a:0:ff:b08d:239d
```

Hint: In this step, you may sometimes get a record in CNAME: you should use the CNAME to continue query and consequently find a record in A type

Another test examples

Results may different every time!

Please NOTE that if the result is read from cache, you should print "read from cache" in the console where your server runs

```
dig @127.0.0.1 www.sina.com a -p 5533

;<<>> DiG 9.16.26 <<>> @127.0.0.1 www.sina.com a -p 5533
;; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15178
;; flags: qr rd ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.sina.com.                IN      A

;; ANSWER SECTION:
www.sina.com.                 60      IN      CNAME   spool.grid.sinaedge.com.
spool.grid.sinaedge.com.     60      IN      CNAME   ww1.sinaimg.cn.w.alikunlun.com.
ww1.sinaimg.cn.w.alikunlun.com. 60      IN      A       120.39.194.227

;; Query time: 1566 msec
;; SERVER: 127.0.0.1#5533(127.0.0.1)
;; WHEN: Sat Mar 05 12:48:10 ;; MSG SIZE rcvd: 121
```



```
dig @127.0.0.1 www.sustech.edu.cn a -p 5533

;<<>> DiG 9.16.26 <<>> @127.0.0.1 www.sustech.edu.cn a -p 5533
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52651
;; flags: qr rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.sustech.edu.cn.                IN      A

;; ANSWER SECTION:
www.sustech.edu.cn.                60      IN      CNAME    www.sustech.edu.cn.w.cdngslb.com.
www.sustech.edu.cn.w.cdngslb.com. 60      IN      A        125.77.142.122

;; Query time: 1748 msec
;; SERVER: 127.0.0.1#5533(127.0.0.1)
;; WHEN: Sat Mar 05 12:51:21 ;; MSG SIZE  rcvd: 98
```

```
dig @127.0.0.1 www.bilibili.com a -p 5533

;<<>> DiG 9.16.26 <<>> @127.0.0.1 www.bilibili.com a -p 5533
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62872
;; flags: qr rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.bilibili.com.                IN      A

;; ANSWER SECTION:
www.bilibili.com.                60      IN      CNAME    s.w.bilicdn1.com.
s.w.bilicdn1.com.                60      IN      A        47.103.24.173

;; Query time: 1565 msec
;; SERVER: 127.0.0.1#5533(127.0.0.1)
;; WHEN: Sat Mar 05 13:07:42 ;; MSG SIZE  rcvd: 77
```

```

>dig @127.0.0.1 www.github.com a -p 5533

;<<>> DiG 9.16.26 <<>> @127.0.0.1 www.github.com a -p 5533
;(1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43045
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.github.com.                IN      A

;; ANSWER SECTION:
www.github.com.                60      IN      A      20.205.243.166

;; Query time: 608 msec
;; SERVER: 127.0.0.1#5533(127.0.0.1)
;; WHEN: Sat Mar 05 12:50:19 ;; MSG SIZE rcvd: 48

```

when read from cache

```

D:\anaconda\envs\py397\python.exe D:/CS305-Project-simple_p2p_simulation/example_complete.py
Enter your ip: 192.168.1.103
Enter your port: 55555
198.41.0.4
a.root-servers.net.
read from cache

```

```

>dig @127.0.0.1 www.baidu.com a -p 5533

;<<>> DiG 9.16.26 <<>> @127.0.0.1 www.baidu.com a -p 5533
;(1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29091
;; flags: qr rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.baidu.com.                IN      A

;; ANSWER SECTION:
www.baidu.com.                60      IN      CNAME  www.a.shifen.com.
www.a.shifen.com.            60      IN      A      14.215.177.38

;; Query time: 0 msec
;; SERVER: 127.0.0.1#5533(127.0.0.1)
;; WHEN: Sat Mar 05 14:11:53 ;; MSG SIZE rcvd: 74

```