The traceroute utility can be used to determine the interim hops leading to a destination address. For example, on a Mac issuing traceroute -I -e www.yahoo.com yields:

(161.44.212.1) 15.145 ms 1.262 ms 1.208 ms
(161.44.206.77) 3.656 ms 1.587 ms 3.001 ms
(161.44.206.29) 2.442 ms 2.486 ms 1.273 ms
(10.112.3.218) 2.600 ms 2.827 ms 2.689 ms
(10.112.8.154) 35.033 ms 35.194 ms 35.186 ms
(10.112.8.30) 35.544 ms 35.486 ms 35.361 ms
(10.112.8.25) 36.341 ms 35.137 ms 35.164 ms
(10.112.2.250) 58.152 ms 58.303 ms 58.548 ms
(10.112.3.58) 58.305 ms 58.039 ms 95.032 ms
(10.81.255.114) 58.118 ms 58.214 ms 58.243 ms
(64.102.241.140) 58.804 ms 58.955 ms 59.005 ms
(64.102.255.153) 59.597 ms 59.510 ms 59.659 ms
(64.102.254.197) 58.780 ms 60.219 ms 58.706 ms
(4.71.162.1) 59.955 ms 59.712 ms 59.926 ms
* * *
(4.14.4.250) 85.400 ms 89.346 ms 86.342 ms
(72.30.223.5) 83.214 ms 83.016 ms 83.691 ms
(72.30.223.27) 83.080 ms 86.814 ms 83.031 ms
(74.6.122.89) 83.041 ms 83.204 ms 83.186 ms
(98.137.192.151) 87.312 ms 90.655 ms 121.103 ms
(72.30.35.10) 86.635 ms 86.754 ms 86.777 ms

Traceroute is able to get the interim hops leading by gradually incrementing the TTL in the IP header. When the TTL reaches 0, the network node in question sends an ICMP to the sender. You can find more details here and here.

Implement a program written in python or C which outputs the common next hops in trying to reach two different hosts. For example, if we take the above output, and the output in trying to reach www.google.com, which is:

(161.44.212.1) 1.856 ms 1.769 ms 1.762 ms
(161.44.206.77) 3.393 ms 3.666 ms 2.984 ms
(161.44.206.29) 2.574 ms 2.333 ms 2.472 ms
(10.112.3.218) 2.544 ms 3.237 ms 3.530 ms
(10.112.8.154) 34.999 ms 33.925 ms 35.064 ms
(10.112.8.14) 35.749 ms 35.852 ms 34.432 ms
(10.112.8.9) 35.090 ms 35.738 ms 35.623 ms
(10.112.2.250) 58.218 ms 58.076 ms 59.050 ms
(10.112.3.58) 58.725 ms 57.811 ms 58.810 ms

(10.81.255.114)  58.829 ms  57.622 ms  58.661 ms
(64.102.241.141)  59.948 ms  58.976 ms  58.822 ms
(64.102.255.149)  59.320 ms  59.498 ms  59.428 ms
(64.102.254.229)  58.888 ms  59.037 ms  59.013 ms
(206.126.236.21)  66.457 ms  71.301 ms  66.401 ms
(108.170.246.33)  67.683 ms  67.617 ms  68.449 ms
(216.239.48.177)  66.623 ms  67.077 ms  66.554 ms
(172.217.7.228)  67.033 ms  66.474 ms  66.867 ms

The program would output the following:

(161.44.212.1)  1.856 ms  1.769 ms  1.762 ms
(161.44.206.77)  3.393 ms  3.666 ms  2.984 ms
(161.44.206.29)  2.574 ms  2.333 ms  2.472 ms
(10.112.3.218)  2.544 ms  3.237 ms  3.530 ms
(10.112.8.154)  34.999 ms  33.925 ms  35.064 ms