Provide 3 subnet addresses in the form \(a.b.c.d/x\) for three subnets:

* Subnet 1, Subnet 2, & Subnet 3

* All subnets must have the prefix \(223.1.13.0/24\)

* Subnet 1 is required to have support for 125 hosts

* Subnets 2 & 3 are required to support 60 hosts

---

Subnet 0: Support 125 hosts

6 bits? \(2^6 - 2 = 62\) (nope!)

7 bits? \(2^7 - 2 = 126\) (yes!)

\(223.1.13.0/24\)

the "/24" means 24 first bits are used in this network's network address, so the first 24 bits cannot be changed within our subnets. This leaves us 8 bits to play with. Of these 8 bits, the last 7 bits we need to distinguish host #’s, which leaves us 1 bit to distinguish this subnet. We can either set it to 0 or 1.

Let's choose 0. Now this subnet uses 25 bits in its network address so it becomes a "/25" network

\((24 \text{ (before)} + 1 \text{ (now)} = 25)\)

Subnet 1: 223.1.13.0/25

\((223.1.13.0 \text{ 0 000 0000})\)

new network address bit host # bits
We could also have chosen to indicate this subnet as

\[
223.1.13. 1 \overset{\wedge}{0} 00000000 \\
\text{new network (address bit)}
\]

which would have given us

Subnet #1: 223.1.13.128/25

Subnets 2 & 3: Support 60

6 bits? \(2^6-2=62\) \(\checkmark\) (yes)

Since the last 6 bits are used for interface numbers and the provider network is a "/24" we have two bits to distinguish our subnets. (Networks will be "/26")

But recall we've already made subnet #1 as (case 1) 223.1.13.0/25 or (case 2) 223.1.13.128/25

Since the 25\(^{th}\) bit as a "0" is indicating subnet 1, we use '10' and '11' to indicate subnets 2 & 3, respectively, for the 25\(^{th}\) and 26\(^{th}\) bits

Subnet 2: 223.1.13. 10 \overset{\wedge}{0} 000000 /26

\text{added network part}

\text{host part}

so 223.1.13.128/26 \(\rightarrow\) subnet 2

Subnet 3: 223.1.13.11 \overset{\wedge}{0} 000000 /26

\text{added network part}

\text{host part}

so 223.1.13.192/26 \(\rightarrow\) subnet 3
<table>
<thead>
<tr>
<th>Subnet</th>
<th>CIDRized Network Address</th>
<th>Broadcast Address</th>
<th># Hosts Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet 1</td>
<td>223.1.13.0/25</td>
<td>223.1.13.127</td>
<td>$2^{2}-2=126$</td>
</tr>
<tr>
<td>Subnet 2</td>
<td>223.1.13.128/26</td>
<td>223.1.13.191</td>
<td>$2^{6}-2=62$</td>
</tr>
<tr>
<td>Subnet 3</td>
<td>223.1.13.192/26</td>
<td>223.1.13.255</td>
<td>$2^{6}-2=62$</td>
</tr>
</tbody>
</table>

Also note:

<table>
<thead>
<tr>
<th>Subnet 1</th>
<th>(sub)Network Mask</th>
<th>Host Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet 1</td>
<td>255.255.255.128</td>
<td>0.0.0.127</td>
</tr>
<tr>
<td>Subnet 2</td>
<td>255.255.255.192</td>
<td>0.0.0.63</td>
</tr>
<tr>
<td>Subnet 3</td>
<td>255.255.255.192</td>
<td>0.0.0.63</td>
</tr>
</tbody>
</table>

Net mask can be logically AND'ed with any host IP in a network to get the network address. Host mask - same method but it extracts the host #.