Fast Subnetting Examples

Remember to always place largest subnets first. How many hosts and how many subnets?

https://www.youtube.com/user/TheLITCNTT

Class C Subnetting - Subnetting for Hosts

You are given 192.168.1.0 to start with, and are being asked for:

- one subnet that contains at least 50 usable IP addresses.

- two subnets that contains at least 14 usable IP addresses each.

Start all this with the biggest subnet. Consider our bit increments:

128 64 32 16 8 4 2 1

We need 50, so draw the line where 50 is:

128 64 **ll** 32 16 8 4 2 1

32+16+8+4+2+1=63 This fits so we use it. *That 64 marks where our first network bit is.*

Our block ends at 192.168.1.63 (our broadcast address) and our first network is 192.168.1.0
What's the subnet mask? Add the unused bits: 128+64=192. The subnet mask is 255.255.255.192
CIDR/ Prefix length? 8 8 8 2 Count all the network bits- 26, 8+8+8+(128 and 64= 2 bits)

So, this subnet is 192.168.1.0/26 Part 1 is done, move on to part 2:

Second subnet starts at 192.168.1.64

We need 14, so draw the line:

128 64 32 16 **ll** 8 4 2 1

8+4+2+1=15 - This fits so we use it. *That 16 marks where our first network bit is.*

64 is this new network number, and with a block of 15, it's broadcast is 192.168.1.79
What's the subnet mask? Add unused bits: 128+64+32+16=240 The subnet mask is 255.255.255.240
CIDR/ Prefix length? 8 8 8 4 Count all the network bits- 28 (128, 64, 32, 16= 4 bits)

So, our second subnet is 192.168.1.64/28 Part 2 is done, and we are almost finished

----

Third subnet starts at 192.168.1.80

We need 14 again:

128 64 32 16 **ll** 8 4 2 1

8+4+2+1=15 - Just like the last step

Our block ends at 192.168.1.95 (our broadcast address)
What's the subnet mask? Add unused bits: 128+64+32+16=240 The subnet mask is 255.255.255.240
CIDR/ Prefix length? 8 8 8 4 Count all the network bits- 28 (128, 64, 32, 16= 4 bits)
So, our third subnet is 192.168.1.80/28, and we are done!
Next block available to continue begins at 192.168.1.96 so you have 96-255 to use in the future (block of 159 numbers for the 192.168.1.0 network we own)

You may be wondering... The line for subnet 1 was drawn between 64 and 32. 128 and 64 were included in counting up network bits when we determine subnet mask and the CIDR, but 64 is the block size, which is the 7th bit, not the 6th. Why?
The reason for that is that is the 7th bit is now designated to chop up into networks, so it's a network bit. It has been appropriated (i.e. "borrowed) from the host bits. The block size is designated by the number to the left of where we draw the line.

<u>Class B Subnetting - Subnetting for Hosts</u>
You are given 170.70.0.0 to start with, and are being asked for:
- *two subnets that contain 1000 usable IP addresses.*
- *one subnet that contains 500 usable IP addresses.*
- *one subnet that contains 100 usable IP addresses.*

[[Before starting, consider this extended binary chart to accommodate the bigger number of hosts:
[[Third octet: 32768 16384 8192 4096 2048 1024 512 256
[[Fourth octet: 128 64 32 16 8 4 2 1

So what we were handed and are starting off with 170.70.0.0, 255.255.0.0
To make the first subnet of 1000 hosts:
32768 16384 8192 4096 2048 1024 **ll** 512 256
[128 64 32 16 8 4 2 1] <---- we can mostly ignore the 4th octet on this step
1023 - This fits so let's use it. *That 1024 marks where our first network bit is.*

This uses 10 host bits (8 bits for the 4th octet + 2 used in the 3rd= 10)
32 bits-10 bits= 22 bits for the network 8 8 6 0 - 170.70.0.0/22
The line was drawn in the 3rd octet going up to the 1023 mark.
/22 has a block size of 4, a subnet of 252 (256-4)
The subnet mask is = 255.255.252.0
Broadcast is 170.70.3.255 - the next network is 170.70.4.0

The second subnet starts at 170.70.4.0.
We need 1000 hosts again- what we just did, so this is easy. Just update the network number with what we already figured out. We know this is a /22 subnet with increments of 4 as the block size.
Subnet mask is 255.255.252.0
Broadcast is 170.70.7.255 - the next network is 170.70.8.0

Third subnet starts at 170.70.8.0
To make the subnet of 500 hosts:
32768 16384 8192 4096 2048 1024 512 **ll** 256
511 - This fits so we use it. This equals 9 bits, leaving 23 remaining (32 bits-9 bits)
CIDR notation? 8 8 7 0 Network bits used- 23 (512 leaves 7 bits in 3rd octet)
170.70.8.0/23 with a block size of 2
Subnet mask is 255.255.254.0
Broadcast is 170.70.9.255 - the next network is 170.70.10.0

Fourth subnet: 170.70.10.0
To make the subnet of 100 hosts:
128 **ll** 64 32 16 8 4 2 1
127 - This fits so we use it. This equals 7 bits, leaving 25 remaining (32 bits-7 bits)
CIDR notation? 8 8 8 1 Count the bits used- 25 (128 leaves 1 bits in 4rd octet)
170.70.10.0/25 - We didn't even have to touch the third octet for this one.
128 is the 25nd bit, thus subnet mask is 255.255.255.128
Broadcast is 170.70.10.127
Next network will start at 170.70.10.128. *170.70.10.128 through 170.70.244.128 are still unused!*

At the beginning, we determined how many bits, which is the first step. Here it was easy- we found where we could fit 50 bits on the bit counting line, but notice that in a way we counted from the right side. When provisioning networks instead of hosts, we have to figure out how many bits will do the job, and then count from the LEFT, adding to the original network numbers, borrowing from host bits.

Class B Subnetting - Subnetting for Networks

We get a class B: 140.78.0.0, 255.255.0.0; asked for 29 networks. Add 2 for NetID and BC is 31

First question! How many bits? 1+2+4+8+16 = 31 = uses 5 bits

Count 5 bits (from the LEFT for networks) and borrow from the host bits to extend network bits:

140.78. 1 1 1 1 1 <= 248
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1
network bits <---- II ----->host bits

Our new starting subnet mask is going to be 255.255.248.0 (the 5 bits)

140.78. 0 0 0 0 1 0 0 0
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our first network - 140.78.8.0

140.78. 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our first network's broadcast- 140.78.15.255

140.78. 0 0 0 1 0 <<————— increment 1 in binary
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our second network - 140.78.16.0

140.78. 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our second network's broadcast- 140.78.23.255

140.78. 0 0 0 1 1 <<————— increment 1 in binary
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our third network - 140.78.24.0

140.78. 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our third network's broadcast- 140.78.31.255

140.78. 0 0 1 0 0 <<————— increment 1 in binary
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our fourth network - 140.78.32.0

140.78. 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

This is our fourth network's broadcast- 140.78.39.255

...And so forth. We begin to see the pattern- multiples of 8, our dividing line gave us the block size

Net ID	**1st address**	**Last address**	**Broadcast**
140.78.8.0	140.78.8.1	140.78.15.254	140.78.15.255
140.78.16.0	140.78.16.1	140.78.16.254	140.78.23.255
140.78.24.0	140.78.24.1	140.78.31.254	140.78.31.255
140.78.32.0	140.78.32.1	140.78.39.254	140.78.39.255

What's the last network you can give out? 140.78.240.0 - as demonstrated below:

140.78. 1 1 1 1 0
128 64 32 16 8 II 4 2 1 . 128 64 32 16 8 4 2 1

It's broadcast is 140.78.247.255

140.78. 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Class B Subnetting - Subnetting for Networks - Example 2

We are given a Class B: 150.9.0.0, BC: 255.255.0.0

We are asked for 10 networks- always add 2 = We really need 12; for networks count from the LEFT

How many bits? 12, so 1100=12 = 4 bits. Count 4 from the left - New magic number is 16

Our new starting subnet mask is going to be 255.255.240.0 (the 5 bits)

150.9. 0 0 0 1

128 64 32 16 **II** 8 4 2 1 . 128 64 32 16 8 4 2 1

This is our first network - 150.9.16 .0

150.9. 1 1 1 1 1 1 1 1

128 64 32 16 **II** 8 4 2 1 . 128 64 32 16 8 4 2 1

This is our first network's broadcast- 150.9.31.255

The last example walked through discovering our networks and we saw the pattern for block of 8 bits repeating. Below it is the same thing using the new info, but the block is 16.

Net ID	**1st address**	**Last address**	**Broadcast**
150.9.16.0	150.9.16.1	150.9.31.254	150.9.31.255
150.9.32.0	150.9.32.1	150.9.47.254	150.9.47.255

150.9.48.0, 150.9.64.0, 150.9.80.0, 150.9.96.0, 150.9.112.0

What if...? Subnet Masks and borrowing from the fourth octet

What if we instead were working with a net of 10 bits? How do we deal with the subnet mask?

150.9. 1 1 1 1 1 1 1 1 1 1

128 64 32 16 8 4 2 1 . 128 64 **II** 32 16 8 4 2 1

Use 64 (the 7th bit), 255.255.255.192 Since we are borrowing from the 4th octet it might look like a class C, but we can ultimate see by the first octet of the address (150) that this is Class B (128 to 191).

Class B Subnetting - Subnetting for Hosts - Review

We are given a Class B: 160.12.0.0, BC: 255.255.0.0 We are asked for 4080 hosts so add 2= 4082

Bits: where is the line drawn? 4096 **^** 2048 1024 512 256 128 64 32 16 8 4 2 1

So we need 12 bits- for hosts count from the RIGHT

160.12. 0 0 0 1

128 64 32 16 **II** 8 4 2 1 . 128 64 32 16 8 4 2 1

Subnet mask is 240 (128+64+32+16) "Magic number" is 16

160.12.16.0… 160.12.31.255

160.12.32.0, 160.12.48.0, 160.12.64.0, etc

Class C Subnetting - Subnetting for Networks - Review

We are given 201.9.6.0 told to make 25 subnets- add 2 and it's 27.

27 = 5 bits - count from the left

201.9.6. 0 0 0 0 1

128 64 32 16 8 **II** 4 2 1

CIDR is /29 and 255.255.255.248 mask

Network IDs are 201.9.6.8, 201.9.6.16, 201.9.6.24, 201.9.6.32, 201.9.6.40 ... etc.

Class C Subnetting - Subnetting for Hosts - Review

We have 195.12.8.0 and need 40 hosts +2 = 42 = 6 bits to hold

This time count FROM THE RIGHT to place the divider:

195.12.8.

128 64 **II** 32 16 8 4 2 1

Subnet mask is 255.255.255.192 (128+64), CIDR is /26

Network IDs 195.12.8.64, 195.12.8.128, 195.12.8.192, etc.

Variable Length Subnet Mask - VLSM #1
Rule 1: Always make networks from highest to lowest in size
Rule 2: Always go from right to left- all we care about is from the host perspective.

We have a starting address of 172.16.0.0
We need 4 subnets, sized with 8000, 1000, 400, and 100 hosts respectively.

8000 hosts- how many bits? 13 Bits:
8192 **^** 4096 2048 1024 512 256 128 64 32 16 8 4 2 1

So we have this 13- start with the originating network ID (172.16.0.0). *Count from the right:*
172.16   0   0   1
          128 64 32 **ll** 16 8 4 2 1 . 128 64 32 16 8 4 2 1
We get 172.16.0.0/19 (8+8+3 = 19), and know the next network ID is 172.16.32.0.
So, the first VLSM block is 172.16.0.0/19, mask is 255.255.224.0, and broadcast of 172.16.31.255

Second network: 1000 hosts. Line is now between 1024 and 512 (10 bits)
Leave the 32 marker from the first network turned on, but move our dividing line to the new spot
172.16   0   0   1   0   0   1
          128 64 32 16 8 4 **ll** 2 1 . 128 64 32 16 8 4 2 1
This is a /22 subnet (8+8+6). 32 + 4 (36) tells us the next network ID
This VLSM block is 172.16.32.0/22, subnet mask is 255.255.252.0, with a broadcast of 172.16.35.255

Third network: 400 hosts - starts at 172.16.36.0 Do the same thing we just did with previous markers.
Line is between 256 and 512
172.16   0   0   1   0   0   1   1
          128 64 32 16 8 4 2 **ll** 1 . 128 64 32 16 8 4 2 1
This is a /23 subnet (8+8+7). We know from 32 + 4 + 2=38 that the next network ID is 172.16.38.0
This VLSM block is 172.16.36.0/23, subnet mask is 255.255.254.0, with a broadcast of 172.16.37.255

Fourth network: 100 hosts - starts at 172.16.38.0 Do the same thing we just did.
The line is between 128 and 64 we are now borrowing host bits from the 4th octet
172.16   0   0   1   0   0   1   1   1
          128 64 32 16 8 4 2 1 . 128 **ll** 64 32 16 8 4 2 1
This is a /25 subnet (8+8+9). 128 is the next network ID
This VLSM block is 172.16.38.0/25, subnet mask is 255.255.128.0, with a broadcast of 172.16.127.255

Hosts	**Net ID**	**Last address**	**Broadcast**
8000	172.16.0.0	172.16.31.254	172.16.31.255
1000	172.16.32.0	172.16.35.254	172.16.35.255
400	172.16.36.0	172.16.37.254	172.16.37.255
100	172.16.38.0	172.16.38.126	172.16.38.127

Next network would begin at 172.16.38.128 for future space

Variable Length Subnet Mask - VLSM #2
140.58.0.0 is our starting IP. We need networks that are big enough to cover these host blocks:
Blocks for 7000, 2500, 1800; and we need 2 blocks of 900, a 500 block, and 2 tiny two-host networks.

First network: 7000 hosts- how many bits? 13 Bits:
8192 **^** 4096 2048 1024 512 256 128 64 32 16 8 4 2 1
So we have this- start with the originating network ID:
140.58 0 0 1
128 64 32 **II** 16 8 4 2 1 . 128 64 32 16 8 4 2 1
This is a /19 subnet (8+8+3). 32 is the next network ID
This VLSM block is 140.58.0.0/19, subnet mask is 255.255.254.0, with a broadcast of 140.58.31.255

Second network: 2500 hosts. Line is now between 2048 and 4096 (12 bits) - keep the old marker!
140.58 0 0 1 1
128 64 32 16 **II** 8 4 2 1 . 128 64 32 16 8 4 2 1
This is a /20 subnet (8+8+4). 32 + 16 (48) is the next network ID
This VLSM block is 140.58.32.0/20, subnet mask is 255.255.240.0, with a broadcast of 140.58.47.255

Third network: 1800 hosts. Line is now between 1024 and 2048(11 bits)

140.58 0 0 1 1 1
128 64 32 16 8 **II** 4 2 1 . 128 64 32 16 8 4 2 1
This is a /21 subnet (8+8+5). 32 + 16 + 8 (56) is the next network ID
This VLSM block is 140.58.48.0/21, subnet mask is 255.255.248.0, with a broadcast of 140.58.55.255

Fourth network: 900 hosts. Line is now between 512 and 1024 (10 bits)
140.58 0 0 1 1 1 1
128 64 32 16 8 4 **II** 2 1 . 128 64 32 16 8 4 2 1
This is a /22 subnet (8+8+6). 32 + 16 + 8 + 4 (60) is the next network ID
This VLSM block is 140.58.56.0/22, subnet mask is 255.255.252.0, with a broadcast of 140.58.59.255

Fifth network: 900 hosts. Line is between 512 and 1024 (10 bits) again.
140.58 0 0 1 1 1 1
128 64 32 16 8 4 **II** 2 1 . 128 64 32 16 8 4 2 1

Ok- we need to draw our line, but it's already at 4. The interval hasn't changed so add a 4 to the network. The number line is just a workspace to help us keep track of things, so it's not a big deal.
This is a /22 subnet (8+8+6). 32 + 16 + 8 + 4 **+ 4** (64) is the next network ID
This VLSM block is 140.58.60.0/22, subnet mask is 255.255.252.0, with a broadcast of 140.58.63.255

<u>Quick break- reassess the number line. It's getting pretty crowded!</u>
172.16 0 0 1 1 1 1
128 64 32 16 8 4 **II** 2 1 . 128 64 32 16 8 4 2 1
We can fix that. Let's move bits around; 32+16+8+4=64 so lets do this to clean up the workspace:
172.16 0 1 0 0 0 0
128 64 32 16 8 4 **II** 2 1 . 128 64 32 16 8 4 2 1

Sixth network: 500 hosts. Line is now between 256 and 512 (9 bits)
172.16   0   1   0   0   0   0   1
          128 64   32   16   8   4   2   II   1 .   128 64 32 16 8 4 2 1
This is a /23 subnet (8+8+7). 64 + 2 (66) is the next network ID
This VLSM block is 140.58.64.0/23, subnet mask is 255.255.254.0, with a broadcast of 140.58.65.255

Seventh network: 2 hosts.
Line is now behind 4 (we need 2 hosts plus BC and NetID), and we have "zoomed in" on the 4th octet.
172.16   0   1   0   0   0   0   1   0   0   0   0   0 0 1
          128 64   32   16   8   4   2   1 .   128 64 32 16 8 4   II   2 1

This is a /30 subnet (8+8+8+6). Now that we are in the 4th octet, 140.58.66.4 is the next network ID
This VLSM block is 140.58.66.0/30, subnet mask is 255.255.255.252, with a broadcast of 140.58.66.3

Eighth network: 2 hosts. Same info as the last network, just update to match.
172.16   0   1   0   0   0   0   1   0   0   0   0   0 0 1
          128 64   32   16   8   4   2   1 .   128 64 32 16 8 4   II   2 1
This is a /30 subnet (8+8+8+6). Now that we are in the 4th octet, 140.58.66.8 is the next network ID
This VLSM block is 140.58.66.4/30, subnet mask is 255.255.255.252, with a broadcast of 140.58.66.7

Hosts	**Net ID**	**Last address**	**Broadcast**
7000	140.58.0.0	140.58.31.254	140.58.31.255
2500	140.58.32.0	140.58.47.254	140.58.47.255
1800	140.58.48.0	140.58.55.254	140.58.55.255
900	140.58.56.0	140.58.59.254	140.58.59.255
900	140.58.60.0	140.58.63.254	140.58.63.255
500	140.58.64.0	140.58.64.254	140.58.64.255
2	140.56.66.0	140.58.66.2	140.58.66.3
2	140.58.66.4	140.58.66.6	140.58.66.7

Next network would begin at 140.58.66.8

The /30 is considered the smallest network you can make to provision a tiny network for isolated connection coupling. In practice, you will hear about /31s working fine for serial connections which need only 2 addresses provisioned for them, but this is NOT recognized by Cisco as working. A /31 may work fine in a production network, but will is guaranteed to be the wrong answer to Cisco.

140.58.0.0, our starting class B with 65,000 plus hosts was too big for one network. For network 1, we needed 7000 provisioned and we got 8190 so there is plenty of room for growth. Network 2 the same thing: we needed 2500 and got a 4094 ceiling for future growth.

All networks have significant overhead, and the only snag might be is if we needed to add routers to the networks with 2 hosts, in which case it is easy to just make some more tiny networks, since all of these networks to handle the enterprise of thousands of hosts barely made a dent in our original Class B address-space.
[the problem with the adding routers thing would be if one of the new routers was connected out to the internet or something]

The previous pages (subnetting for hosts, subnetting for networks, VLSM 1 & 2) all were made out of the information presented in the classroom videos available here:
https://www.youtube.com/user/TheLITCNTT

Here is the walkthrough of subnet video #8 on the Cisco Press Official ICND1 Guide companion CD
You are given this network number, and you need to plot out the subnets:
10.0.0.0 - 255.255.192.0 /18

1: Note the class of the network address: Class A
2: Note the CIDR - The subnet mask with the class of the network number shows that this subnet is unusual. This exercise will show the importance of the CIDR over the dotted decimal.

What do we know? As a rule Class A uses 8 network bits. The CIDR says 18 network bits.
18-8=10, so we have 10 subnet bits. 32-18 leaves 14 host bits.

Dissect that CIDR to see it in binary:
11111111.11111111.11000000.00000000
Hey! check it out- we have our Class A's 8 binary 1's, and the 2nd and third octet for the 10- plus it matches the dotted decimal right? See the 192 in the 3rd octet? Alright!
Wait- it's still confusing. Well, let's forget that for a minute and do this like the Cisco videos do all of theirs:

Here's what they've got:
255.255.192.0 - We are going to note the octet that isn't a 0 or 255.
Now 255-192 = 64 for the block size or what they call a "magic number"
10.0.0.0, 10.0.64.0, 10.0.128.0, 10.0.192.0, and 10.0.255.254 must be our broadcast in the last subnet because...NO - remember that we have that whole 2nd octet to fill? Get at it!
Ok start again: 10.0.0.0, 10.0.64.0, 10.0.128.0, 10.0.192.0, 10.1.0.0, 10.1.64.0, 10.1.128.0, 10.1.192.0, 10.2.0.0, 10.2.64.0, 10.2.128.0, 10.2.192.0, and so forth.

According to Cisco, 10.255.192.0 is what they call our "broadcast subnet", our last subnet. We can then extrapolate that our real broadcast host address in that last subnet will then be 10.255.192.255, and that these are the ranges of networks:

Network#	1st host	Last host	Broadcast
10.0.0.0	10.0.0.1	10.0.63.254	10.0.64.255
10.0.64.0	10.0.64.1	10.0.127.254	10.0.128.255
10.0.128.0	10.0.128.1	10.0.191.254	10.0.191.255
10.0.192.0	10.0.192.1	10.0.255.254	10.0.255.255
10.1.0.0			

So what else can we find out about this network?

It looks like [(64 x 256) - 2 = 16382] will tell us the number of hosts in each network and [2^14 - 2] gives us the same answer.
This shows us that in 2^n-2, n only refers to the "borrowed bits" which in this case means anything not accounted for by the Classful system (eight for a Class A), which is 10 of the /18.

It also looks like 256 x 4 = 1024 should tell us how many networks were made, and sure enough, 2^10 is 1024

Subnet Shortcuts
Using binary AND: 1 and 1 = 1, all others = 0 (see below)

What is the network number of the address 192.168.100.115 with a subnet mask of 255.255.255.240?
192.168.100.115 = 11000000.10101000.01100100.01110011
255.255.255.240 = 11111111.11111111.11111111.11110000
ANDed result = 11000000.10101000.01100100.01110000 = 192.168.100.112
192.168.100.115 belongs to the 192.168.100.112 network when a mask of 255.255.255.240 is used.

It get's simpler: octets of the addresses with a common 255 or 0 can be ignored
What's the broadcast for the IP address 192.168.100.164 if it has a subnet mask of 255.255.255.248?
192.168.100.164 = 192.168.100.10100100
255.255.255.248 = ...11111000
ANDed result = 192.168.100.10100000 = 192.168.100.160 (here's our subnet number)

Separate the network bits from the host bits:
255.255.255.248 = /29 = the first 29 bits are network bits - so the last three are host bits
----.10100 **000**. Change all host bits to 1, so it's ----.10100111 = 192.168.100.167
The broadcast of 192.168.100.164 is 192.168.100.167 when the subnet mask is 255.255.255.248.

To what network does 131.186.227.43 belong, if its subnet mask is 255.255.240.0?
Based on the two shortcut rules, the answer should be 131.186.???.0
So now you only need to convert one octet to binary for the ANDing process:
227 = 11100011
240 = 11110000
11100000 = 224 Therefore, the answer is 131.186.224.0

What subnet does this belong to?

What subnet does 192.168.12.78/29 belong to?
Our mask is a /29. The next boundary is 32. So 32-29=3.
Now 2^3 = 8 which gives us our block size i.e. 2 to the power of 3 equals 8.

We have borrowed from the last octet as the 29th bit is in the last octet. We start from zero and count up in our block size. Therefore it follows that the subnets are:
192.168.12.0
192.168.12.8
192.168.12.16
...
192.168.12.72
192.168.12.80
Our address is 192.168.12.78 so it must sit on the 192.168.12.72 subnet.

What subnet does 172.16.116.4/19 sit on?
Our mask is /19 and our next boundary is 24. Therefore 24-19=5. The block size is 2^5 = 32.
We have borrowed into the third octet as bit 19 is in the third octet so we count up our block size in that octet. The subnets are:
172.16.0.0
172.16.32.0
172.16.64.0
172.16.96.0
172.16.128.0
Our address is 172.16.116.4 so it must sit on the 172.16.96.0 subnet.
What subnet does 10.34.67.234/12 sit on?
Our mask is 12. Our next boundary is 16. Therefore 16-12=4. 2^4=16 which gives us our block size.
We have borrowed from the second octet as bit 12 sits in the second octet so we count up the block size in that octet. The subnets are:
10.0.0.0
10.16.0.0
10.32.0.0
10.48.0.0
............etc
Our address is 10.34.67.234 which must sit on the 10.32.0.0 subnet.

What is the *valid* host range of the Nth subnet of abc.xxx.yyy.zzz/xx?

What is the valid host range of the 4th subnet of 192.168.10.0/28?
The block size is 16 since 32-28=4 and 2^4 = 16.
We need to count up in the block size in the last octet as bit 28 is in the last octet.

192.168.10.0
192.168.10.16
192.168.10.32
192.168.10.48
192.168.10.64
The 4th subnet is 192.168.10.48 and the host range must be 192.168.10.49 to 192.168.10.62, remembering that the subnet and broadcast address cannot be used.

What is the valid host range of the 1st subnet of 172.16.0.0/17?
The block size is 128 since 24-17=7 and 2^7 = 128.
We are borrowing in the 3rd octet as bit 17 is in the 3rd octet. Our subnets are:

172.16.0.0
172.16.128.0

The first subnet is 172.16.0.0 and the valid host range is 172.16.0.1 to 172.16.127.254. You must remember not to include the subnet address (172.16.0.0) and the broadcast address (172.16.127.255).

What is the valid host range of the 7th subnet of address 10.0.0.0/14?
The block size is 4, from 16-14=2 then 2^2 = 4.
We are borrowing in the 2nd octet, so count in the block size from zero 7 times to get the 7th subnet.

The seventh subnet is 10.24.0.0. Our valid host range must be 10.24.0.1 to 10.27.255.254 again remembering not to include our subnet (10.24.0.0) and the broadcast address (10.27.255.255).

10.0.0.0
10.4.0.0.0
10.8.0.0.0
...
10.20.0.0.0
10.24.0.0.0

Dotted decimal mask? What's the block size?
A mask of 255.255.192.0 - you would simply count up in 256-192 = 64 in the third octet.
Or 255.224.0.0 - block size is 256-224=32 in the second octet.

Question: You are designing a subnet mask for the 10.0.0.0 network. You want 3800 subnets with up to 3800 hosts on each subnet. What subnet mask should you use?
Answer: 255.255.240.0

Question: How many subnets and hosts per subnet can you get from the network 172.29.0.0 255.255.254.0?
Answer: 128 subnets and 510 hosts

Just memorize how many items you get for each number of bits (subtract two for hosts)

000000000001 = 2 (1)
000000000011 = 4 (2)
000000000111 = 8 (3)
000000001111 = 16 (4)
000000011111 = 32 (5)
000000111111 = 64 (6)
000001111111 = 128 (7)
000011111111 = 256 (8)
000111111111 = 512 (9)
001111111111 = 1024 (10)
011111111111 = 2048 (11)
111111111111 = 4096 (12)

If you can't remember what one is, just remember it's twice the number of the one before it (or halve the number after it if you're counting down). There's no math, tricks, etc. involved.

So for the first question, you'll need 12 bits for both because the value is between 2048 and 4096 (you obviously have to go with the higher value).

You can look at the other question the same way. /23 on a class B will give you 7 subnet bits and 9 host bits. 7 bits is 128 and 9 bits is 512 (minus 2 gives you 510).

Different answer:

Bits do you need to borrow to accommodate 3800 subnets?
12 bits as 2 ^ 12 = 4096.
Your network address is a /8 by default as it is a Class A address so 8 + 12 = /20 mask.

The second question is straightforward.
Your address is Class B so it has a default mask of /16.
255.255.254.0 is the same as /23, so 23-16=7.
2^7 = 128 subnets
(2 ^ (32 - 23))-2 = 2^9-2 = 510 hosts.

Interesting:
Question: *What is the last valid host on the subnetwork 172.25.248.0/21?*
Answer: 172.25.255.254

1st question: what am I being asked for? Subnets or hosts (or both)
Add 2 - for all 1's and all 0's addresses - for networks this gets iffy

Subnet mask cheat 128+64=192, 192+32=224, 224+16=240, etc:

	192	**224**	**240**	**248**	**252**	**254**	**255**
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**

Side Note: patterns in binary- in sequence, the numbers repeat and alternate in their own increments as shown below

4	**2**	**1**
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Net ID	**1st address**	**Last address**	**Broadcast**
195.12.8.64	195.12.8.65	195.12.8.126	195.12.8.127
195.12.8.128	195.12.8.129	195.12.8.190	195.12.8.191

YOU CANT USE 192.12.8.192 IT CONFLICTS WITH SUBNET MASK??
SAID 2 times in video!

Host = 2^n-2   2^(number of host bits) -2 --- 2^6-2= 64-2= 62 hosts
2 bits in the network portion, 2^n-2 is 4-2= 2 and we have two networks.

How many networks? 2^5 - 2 networks = 32-2= 30

128	64	32	16	8	4	2	1
2^7	2^6	2^5th	2^4	2^3	2^2	2^1	2^0

How many hosts per network? 2^3 - 2 hosts = 8-2= 6

128	64	32	16	8	4	2	1
2^7	2^6	2^5th	2^4	2^3	2^2	2^1	2^0

Steps

1- How many networks or hosts do you need
2- Convert it to binary- how many bit does it take?
3- Write out the number line
4- If it's a network, count of from the LEFT, if it's hosts, count from the RIGHT to draw your line.
5- Mark your "magic number" and set your first IP address