2.

Solution:

I'll first prove the second part:

Let X $\leq_p$ Y, and Y $\in$ coNP.

From the definition of polynomial time reduction we know that since X is polynomial time reducible to Y, there is an algorithm for transforming an instances of X into instances of Y.

In other words, there is a polynomial reduction function f such that

$\alpha \in$ X $\iff$ f(α) $\in$ Y.

Now since Y is in coNP and X $\leq_p$ Y, we can find a coNP algorithm for X by finding f(α) for any α in X and then applying the coNP algorithm of Y on it. Therefore we can find a coNP algorithm for X using the coNP algorithm we have for Y and X is in coNP.

Proof of the first part:

Let problem P be a NP-hard problem in coNP and problem M be any problem in NP.

By the definition of NP-hard, we can say that every problem in NP poly-time reduces to P. So M polynomial reduces to P.

So now we have M $\leq_p$ P, and P $\in$ coNP.

From the proof above, if we substitute M for X and P for Y, we can conclude that M is also in coNP.

Since M is any problem in NP and M is also in coNP, we can say that any problem is NP is also in coNP or NP is a subset of coNP.