1. Formulate Church-Turing thesis. Is it a mathematical theorem? A statement about the physical world?

Anything that can be computed on any physical device can also be computed using a turing-machine or Java program.

This is not a mathematical theorem, it is a statement about the physical world.
2. Formulate the current definition of a feasible algorithm. Give two examples explaining why this definition is not perfect:

- an example of an algorithm that is feasible according to this definition but not feasible according to common sense; and
- an example of an algorithm that is feasible from the practical viewpoint but not feasible according to this definition.

An algorithm $A$ is said to be feasible if there exists a polynomial $P$ in which for every input $x$, the run-time $t_A(x)$ of the algorithm is bounded by $P(len(x))$; $t_A(x) \leq P(len(x))$

1. $t_A(x) = 10^{100} \times len(x)$
   - Feasible according to the definition
   - Not feasible to common sense

2. $t_A(x) = 2^{10^{-12}} \times len(x)$
   - Not feasible according to the definition
   - Feasible in practice
3. Define what is a problem from the class NP.

NP we have

1) a feasible algorithm \( c(x, y) \) that, given strings \( x \) and \( y \), returns true or false

2) a polynomial \( p(n) \)

Given \( x \)

Find \( y \) such that

\( c(x, y) \) is true

and \( \text{len}(y) \leq p(|x|) \)

A problem is said to be NP-hard if there exists an algorithm for solving it that can be translated into any other NP problem. In other words, all NP problems can be reduced to this problem.

An example would be the knapsack problem because the only way to do it is by checking every option available.
5. Can every NP-hard problem be solved by an algorithm? Can every NP-hard problem be solved by a feasible algorithm?

Yes every NP-hard problem can be solved by an algorithm but what is not known is that if it can be solved by using a feasible algorithm. It is not known if every NP-hard or every NP problem can be solved by a feasible algorithm, it is still an open problem.