1. **What is a P, an NP, an NPC or an NP-Hard problem? Give at least one example for each type?**

**P** problems are decision problems for which we can find an efficient algorithm to solve said problem at worst polynomial time (meaning at most n ^ O(1) ).

**NP** problems are decision problems where there exist solutions that can be verified by a non-deterministic polynomial time algorithm. This is to say that it is not a solution to the problem, but rather an algorithm that given a “YES” answer to our decision problem, it can prove and check the proof in polynomial time.

**NPC (NP Complete)** problems are that are within the of **NP** and **NP-Hard.**

**NP-Hard** problems are NP problems that are **at least** as hard as the hardest problems in NP. This claims that the set of NP-Hard problems are at the edge or beyond in terms of complexity difficulty in a set ordered by difficulty. So, NP-Hard problems could be outside of the NP set, into **EXP** problems or beyond.

1. **What are the relationships among P, NP, NPC and NP-hard complexity classes?**

All **P** problems are **NP** problems. **NP-Hard** are at least within the hardest problems of **NP.**

**NPC** problems are those which are in **NP** and in **NP-Hard**.