## P(Polynomial) Vs NP (Non Polynomial)

**Name:Ashwin Shiv**

**Roll No:181210013**

**CSE 2nd Year**

## Polynomial Time Algorithm Vs Non-Polynomial Time Algorithm:

## Polynomial Time Algorithm:

### **Algorithm whose time complexity is in a form of a polynomial.**

### **Eg:Linear Search:O(n),Binary Search:O(logn),Matrix Multlipication:O(n3).**

## Non-Polynomial Time Algorithm:

### **Algorithm whose time complexity is in a form of a non-polynomial form.**

### **Eg:0/1 Knapsack Problem:O(2n),Travelling Salesman Problem:O(2n).All Non-Polynomial Time Algorithms are NP-Hard.**

# **Deterministic Algorithm Vs Non- Deterministic Algorithm**:

## Deterministic Algorithm:

### **For a particular input the computer will give always same output.**

### **Can solve the problem in polynomial time.**

## Non-Deterministic Algorithm:

### **For a particular input the computer will give different output on different execution.**

### **Can’t solve the problem in polynomial time.**

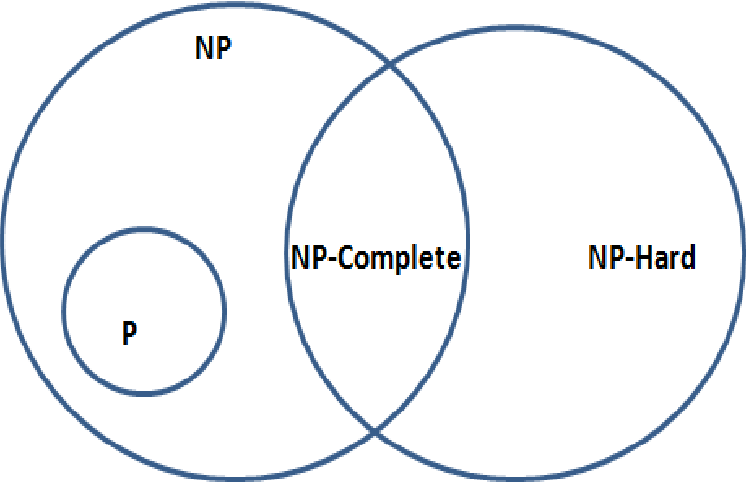
#### **Features of Non-Deterministic Algorithm:**

### **choice(X) : chooses any value randomly from the set X.**

### **failure() : denotes the unsuccessful solution.**

### **success() : Solution is successful and current thread terminates.**

### **P vs NP vs NP-Hard vs NP-Complete**



### **Difference between P and NP**

|  |  |
| --- | --- |
| **P(Polynomial)** | **NP(Non-Polynomial)** |
| 1. Problems which can be solved in polynomial time by deterministic algorithms. | 1. Problems which cannot be solved in non- deterministic polynomial time. |
| 1. Can be solved and verified in polynomial time. | 1. Cannot be obtained in polynomial time but if solution is given, it can be verified in polynomial time |
| 1. Subsets of NP Problems. | 1. Supersets of P Problems. |
| 1. They are deterministic in nature | 1. They are non-deterministic in nature |
| 1. Example: Linear Search, Selection Sort, Merge Sort, Matrix Multiplication etc. | 1. Example: Travelling Salesman Problem,0/1 Knapsack Problem etc. |