UNIT 5

Tractable and Untractable Problems -

1. **Tractable Problems** -
   
   The set of all the problems that can be solved within polynomial amount of time using deterministic machine.

2. **Untractable Problems** -
   
   The set of all the problems that can't be solved within polynomial amount of time using deterministic machine.

3. **P class problem** -
   
   A language \( L \) is in class \( P \) if there exists some polynomial \( T(n) \) such that \( L = T(M) \) for some deterministic TM \( M \) of time complexity \( T(n) \).

4. **NP class problem** -
   
   A language \( L \) is in class \( NP \) if there is a non-deterministic TM \( M \) and a polynomial time complexity \( T(n) \) such that \( L = T(M) \) and \( M \) executes at most \( T(n) \) moves for every input \( w \) of length \( n \).

5. **Polynomial time reduction** -
   
   Let \( P_1 \) and \( P_2 \) be two problems. A reduction from \( P_1 \) to \( P_2 \) is an algorithm which converts an instance of \( P_1 \) to an instance of \( P_2 \). If the time taken by the algorithm is a polynomial \( p(n) \), \( n \) being the length of the input of \( P_1 \), then the reduction is called a polynomial time reduction \( P_1 \) to \( P_2 \).

   \[ \rightarrow \] If there is a polynomial time reduction from \( P_1 \) to \( P_2 \) and if \( P_2 \) is in \( P \) then \( P_1 \) is in \( P \).

6. **NP-complete problem** -
   
   Let \( L \) be a language or problem in \( NP \). The \( L \) is NP-complete if
   
   (i) \( L \) is in \( NP \)
   
   (ii) \( L \) is an NP-complete if
   
   (iii) Every language \( L' \) in \( NP \) there exists a polynomial time reduction of \( L' \) to \( L \).
→ If $P_1$ is NP-complete, and there is a polynomial-time reduction of $P_1$ to $P_2$, then $P_2$ is NP-complete.

→ If some NP-complete problem is in P, then P = NP.

5. **NP-hard problems** -

These problems are at least as hard as the hardest problem in NP but not necessarily in NP.

The problem to which all NP-class problems are reducible in polynomial time is known as NP-hard problems.

6. **NP-complete problems** -

(i) SAT problem (satisfiability problem for boolean expressions)
(ii) Hamiltonian Path Problem (HPP)
(iii) Travelling Salesman Problem (TSP)
(iv) Vertex Cover Problem (VCP)
(v) Partition problem (PP)

Cook's theorem $\rightarrow$ SAT is NP-complete.