

**KONERU LAKSHMAIAH EDUCATION FOUNDATION**

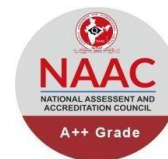
(Deemed to be University estd, u/s, 3 of the UGC Act, 1956)

(NAAC Accredited “A++” Grade University)

Green Fields, Guntur District, A.P., India – 522502

Department of Computer Science and Engineering

(DST - FIST Sponsored Department)

**Active Learning Method**

Program: B. Tech – CSE

Academic Year / Yr-Sem : 2024 - 25 / II - II Sem

Course Title & Code: DESIGN AND ANALYSIS OF ALGORITHMS – 22CS2205R

CO#	4
Topics	NP-Hard Graph Problem
Type of ALM	DEBATE
Learning Approach	EXPERENTIAL LEARNING

Q1: How to find CDP Problem is NP-Hard Problem? If the problem is NP-Hard then convert the exponential time complexity to the polynomial time complexity. Also compare the deterministic and non-deterministic algorithms with polynomial time execution.

◆ Is CDP NP-Hard?

To prove CDP (Combinatorial Decision Problem) is **NP-Hard**:

- Take a known NP-Hard problem (e.g., 3-SAT or Subset Sum).
- Reduce it to CDP in **polynomial time**.
- If such a reduction exists, CDP is NP-Hard.

◆ Can we convert exponential time to polynomial time?

- **Not generally**, unless $P = NP$ (still unsolved).
- Use:
 - **Dynamic Programming** (for pseudo-polynomial time).
 - **Approximations/heuristics** (for near-optimal solutions).

◆ Deterministic vs Non-Deterministic Algorithms

Property	Deterministic	Non-Deterministic
Execution	One path	Tries all paths (theoretical)
Time (for NP)	Exponential (usually)	Polynomial (guess + verify)
Real-world	Yes	No (concept only)

Q2: How to find NCDP Problem is NP-Hard Problem? If the problem is NP-Hard, then convert the exponential time complexity to the polynomial time complexity. Also compare the deterministic and non-deterministic algorithms with polynomial time execution

◆ How to prove NCDP is NP-Hard?

- Reduce a known NP-Hard problem (like 3-SAT) to NCDP in polynomial time.
- If successful, NCDP is NP-Hard.

◆ Can exponential time be converted to polynomial?

- Not generally, unless $P = NP$.
- Use dynamic programming, approximations, or parameterized algorithms to reduce time in practice.

◆ Deterministic vs Non-Deterministic Algorithms

Type	Deterministic	Non-Deterministic
Execution	One fixed path	Tries all paths (theoretical)
Time for NP	Exponential	Polynomial (guess + check)
Real-world use	Yes	No (conceptual only)