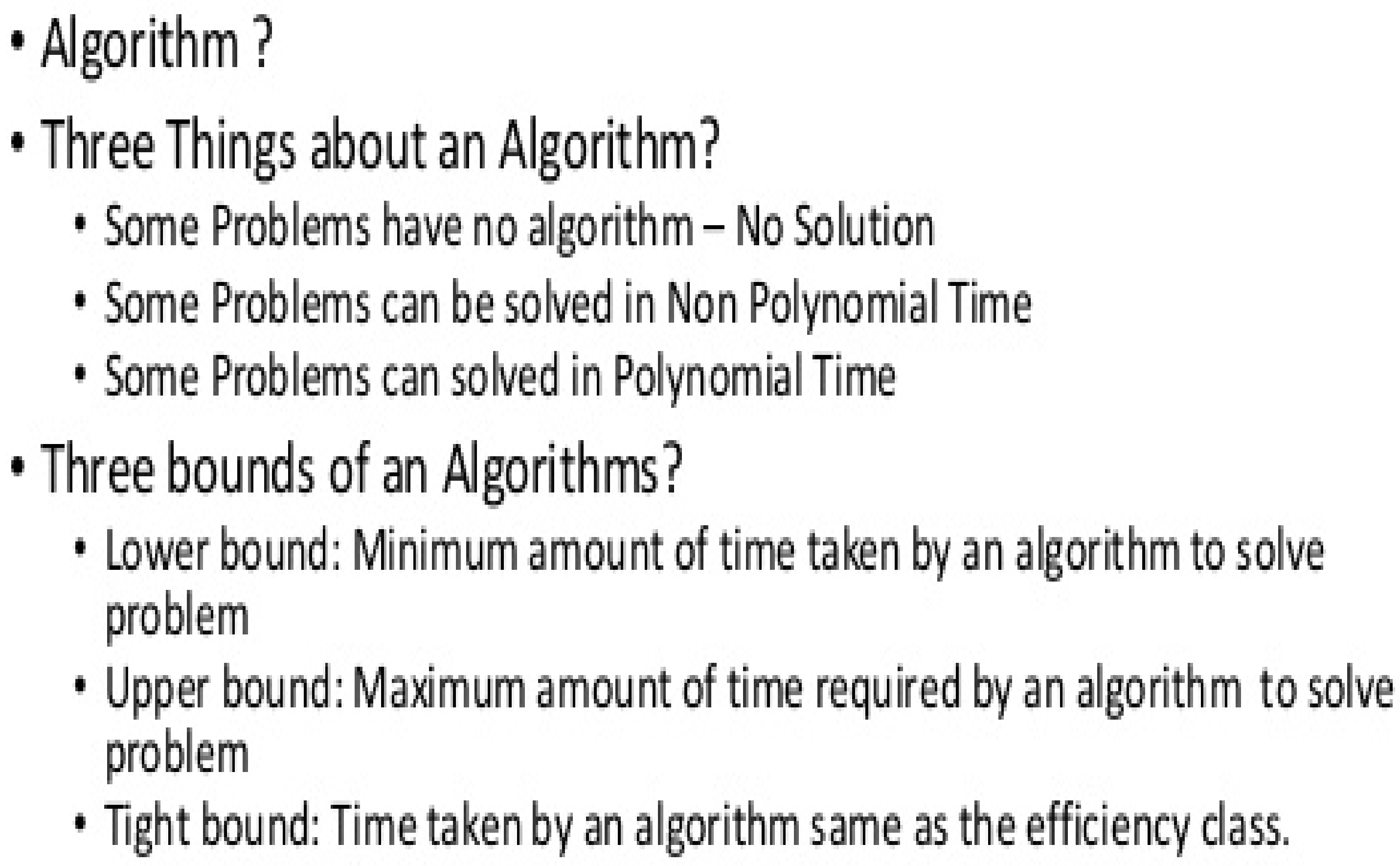
SNS COLLEGE OF TECHNOLOGY, COIMBATORE –35

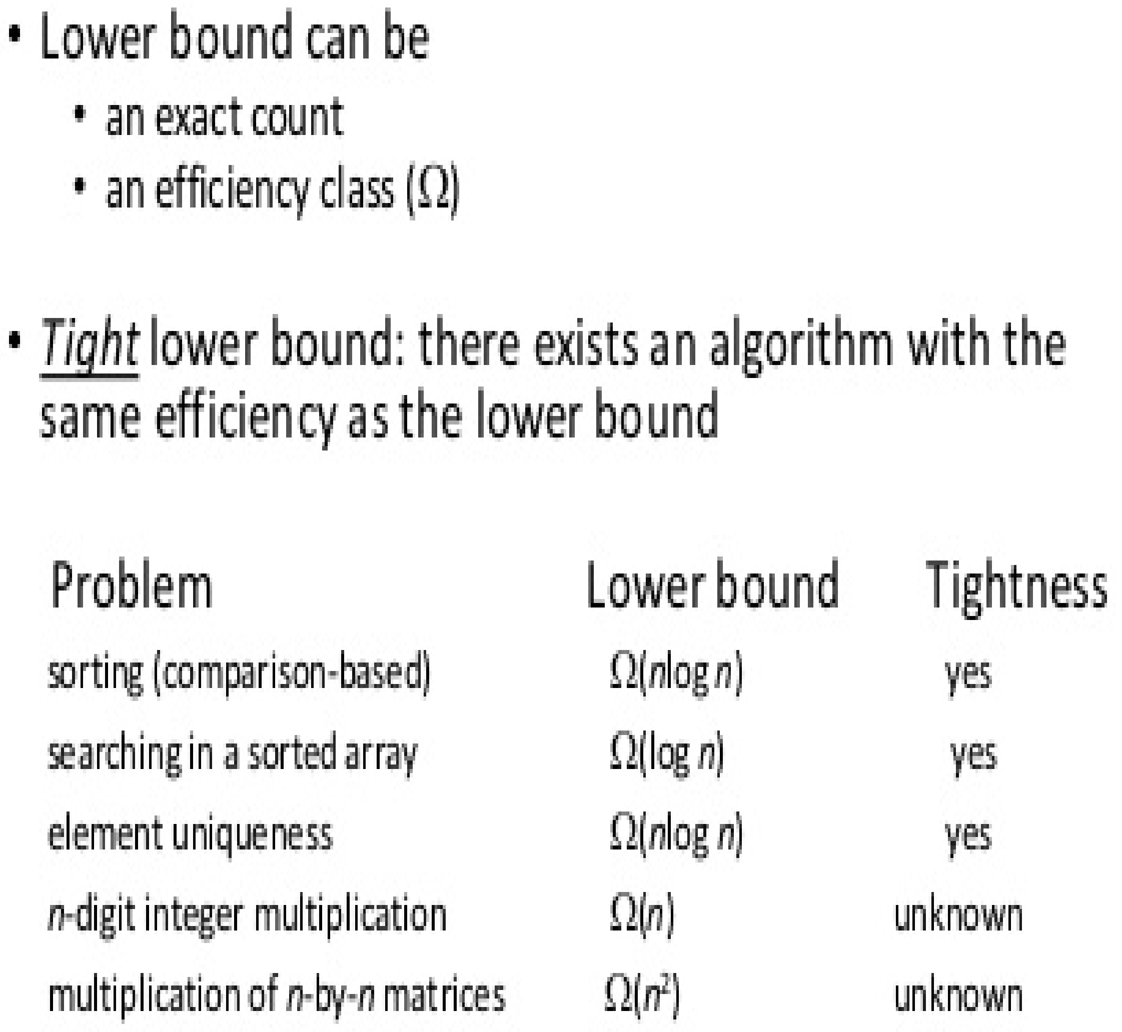
Limitations of Algorithm, Lower Bound Arguments

Limitations of Algorithm



- Algorithm ?
- Three Things about an Algorithm?
  - Some Problems have no algorithm – No Solution
  - Some Problems can be solved in Non Polynomial Time
  - Some Problems can solved in Polynomial Time
- Three bounds of an Algorithms?
  - Lower bound: Minimum amount of time taken by an algorithm to solve problem
  - Upper bound: Maximum amount of time required by an algorithm to solve problem
  - Tight bound: Time taken by an algorithm same as the efficiency class.

Lower Bound Arguments

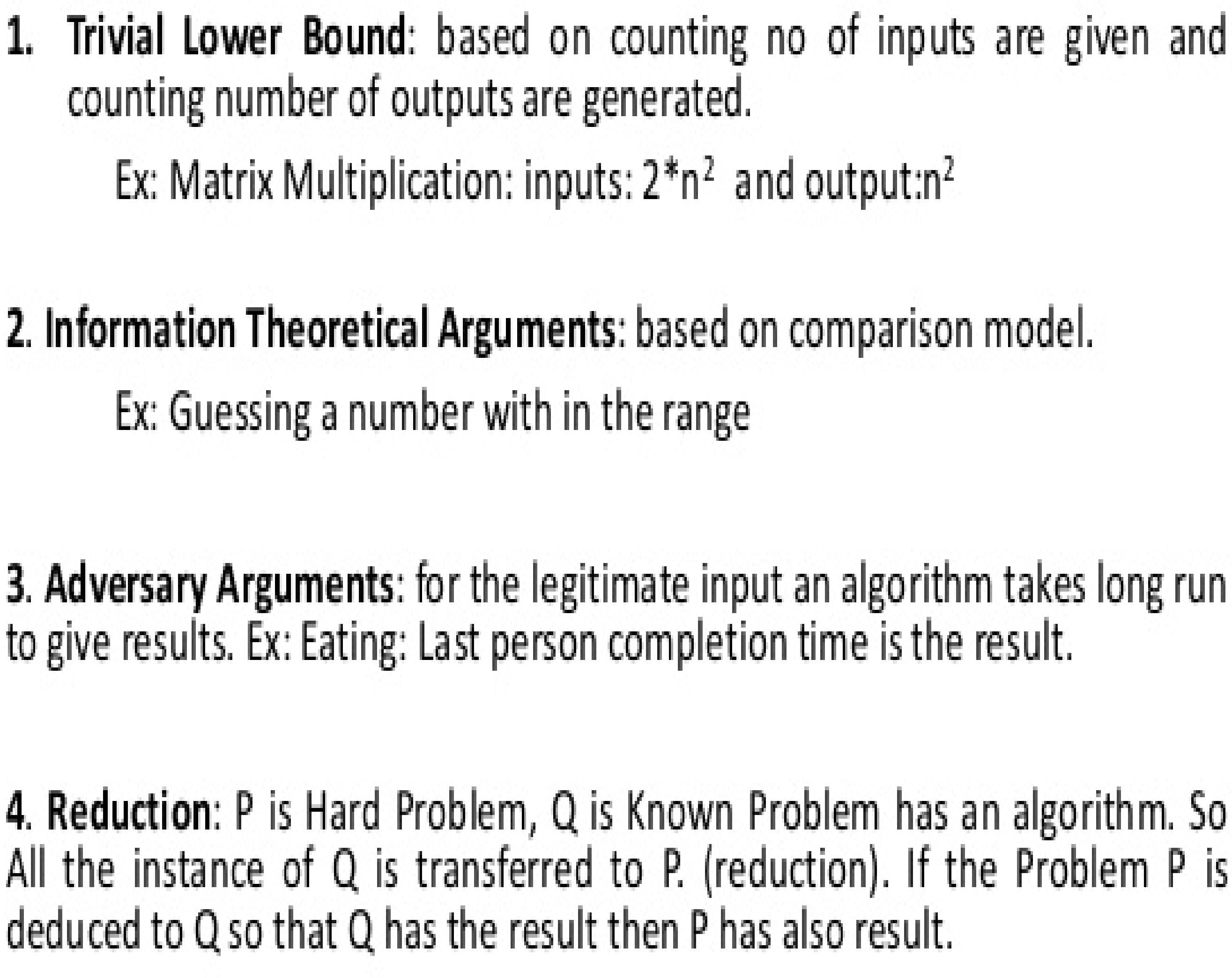


- Lower bound can be
  - an exact count
  - an efficiency class (Ω)

- *Tight* lower bound: there exists an algorithm with the same efficiency as the lower bound

Problem	Lower bound	Tightness
sorting (comparison-based)	$\Omega(n\log n)$	yes
searching in a sorted array	$\Omega(\log n)$	yes
element uniqueness	$\Omega(n\log n)$	yes
n-digit integer multiplication	$\Omega(n)$	unknown
multiplication of n-by-n matrices	$\Omega(n^2)$	unknown

Four types of Obtaining Algorithm Lower Bound



1. **Trivial Lower Bound**: based on counting no of inputs are given and counting number of outputs are generated.

   Ex: Matrix Multiplication: inputs: $2*n^2$ and output:n^2

2. **Information Theoretical Arguments**: based on comparison model.

   Ex: Guessing a number with in the range

3. **Adversary Arguments**: for the legitimate input an algorithm takes long run to give results. Ex: Eating: Last person completion time is the result.

4. **Reduction**: P is Hard Problem, Q is Known Problem has an algorithm. So All the instance of Q is transferred to P. (reduction). If the Problem P is deduced to Q so that Q has the result then P has also result.