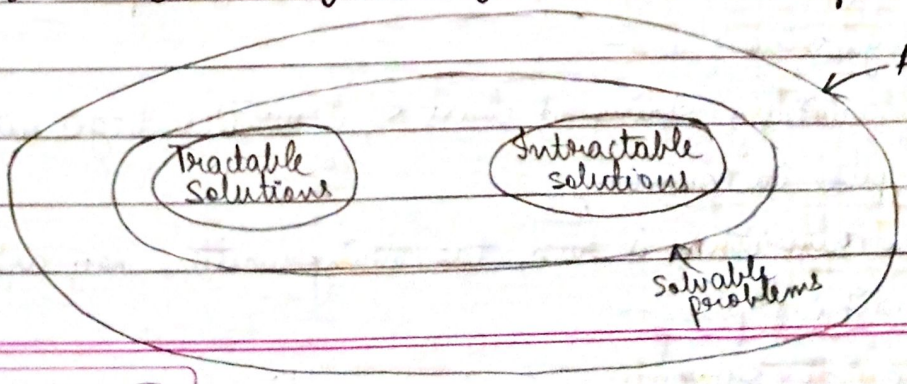
20.04.2020

20. Tractable and Intractable Problems

- An algorithm for a given problem is said to be a polynomial time algorithm if its worst case complexity belongs to $O(n^k)$ for a fixed integer k and an input of size n.
- The set of all problems that can be solved in polynomial amt of time are called Tractable problems. these problems can run in a reasonable amt of time even for very large amounts of input data.
- Intractable problems
  - the set of all problems that can't be solved in polynomial time
  - Uts of type $O(k^n)$.
  - Require huge amts of time for even modest input sizes.



DATE:
PAGE NO.

Deterministic machines	Non-deterministic machines
- Conventional digital machines are deterministic in nature.	- Hypothetical machine.
- Serialization of resource access.	-> 1 job can be done in 1 unit of time.
- Do a sequential exec^n based on: • von Neumann Architecture. • Serializ^n of resource access.	- There will be only 1 hypothetical processor which can do >1 job at any instance of time.
eg: Linear search → O(n)	eg: Linear search → O(1) (parallel fashion)

> P Class / Polynomial Class Problems

The class of decision problems that can be solved in polynomial time by deterministic algorithms.

eg: O(1) – Constant
O(log n) – Sub-linear
O(n) – Linear
O(nlog n) – Nearly linear
O(n²) – Quadratic.

Decision problems are problems with yes/no answers.

> NP Class / Non-deterministic Polynomial Class Problems

The class of decision problems that can be solved in polynomial time by non-deterministic algorithms.

eg:- Graph coloring
- Hamiltonian cycle
- " path
- Job scheduling with penalties
- Bin packing
- Subset sum problem
- Satisfiability "
- Traveling sales person problem.

Non-deterministic Algorithm has 2 phases and 1 o/p step:

(i) Non-deterministic guessing phase -
- Some completely arbitrary string of chars s, is written beginning at some designated place in memory.
- Each time the algorithm is run, the string written may differ.

(ii) Deterministic verifying phase -
- A deterministic subroutine begins execution.
- In addition to the decision problem's input, the subroutine may use or ignore s.

- Eventually, it returns a value true or false - or it may get in an infinite loop and never halt.

(iii) Output step - If the verifying phase returned true, the algorithm outputs yes.