**Algorithms**

* A polynomial algorithm is one whose time complexity grows as a polynomial function
* NP-complete: A decision problem Q is said to be NP complete if **Q elemt of NP** and Q**` elemt of NP** and **Q` is transformable to Q**
* Np-hard: A decision problem Q is said to be NP hard if **all problems in NP are polynomially transformable to Q**, but **cannot** show that **Q element of NP**
* One of the **research objectives** in real-time scheduling is to identify simpler, but still practical, **problems that can be solved in polynomial time.**
* To simplify the complexity of a feasible schedule.
  + uniprocessor systems
  + adopting preemptive models
  + using static priorities
  + removing resource constraints
  + using homogeneous task sets
* **Preemptive VS Non preemptive**
  + In preemptive algorithms, the running task can be interrupted at any time to assign the processor to another active task.
  + In non-preemptive algorithms, a task, once started, is executed by the processor until completion
* **Static vs Dynammic**
* **Online vs offline**
* **Optimal vs heuristic**
* If Γ is the current task set that has been previously guaranteed, a newly arrived task τ new is accepted into the system if and only if the task set Γ # = Γ ∪ {τ new } is found schedulable.
* The benefit of having a **guarantee mechanism** is that potential overload situations can be detected in advance to **avoid negative effects on the system**.
* One of the most dangerous phenomena caused by a transient overload is called **domino effect.** It refers to the situation in which the arrival of a new task causes all previously guaranteed tasks to **miss their deadlines**.