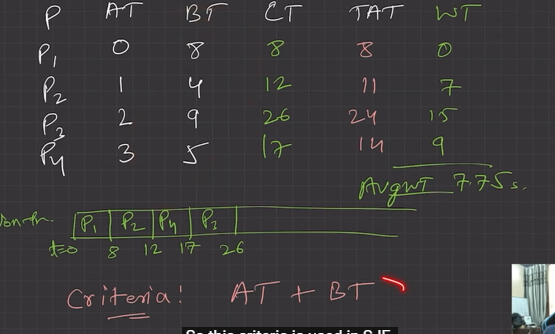
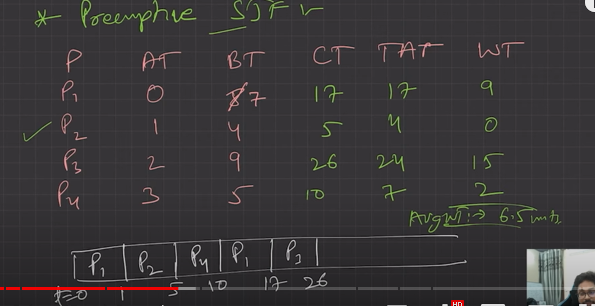
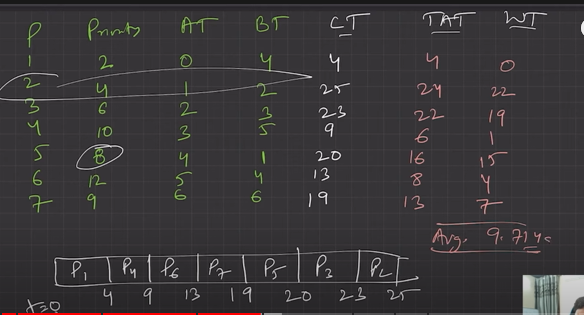
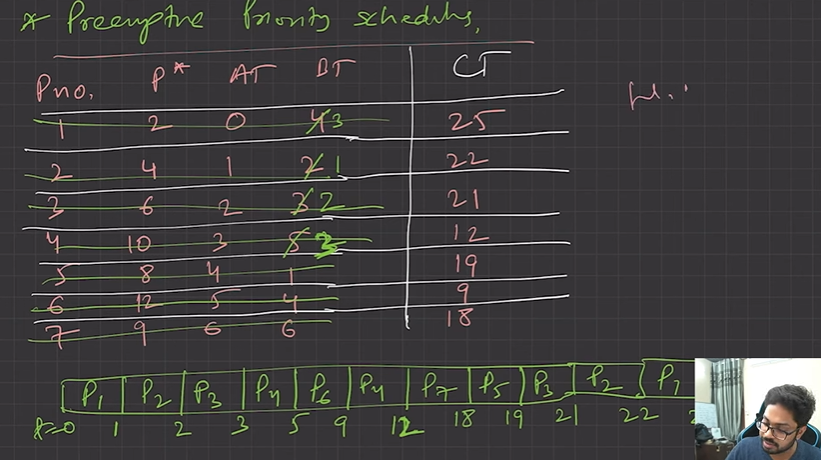
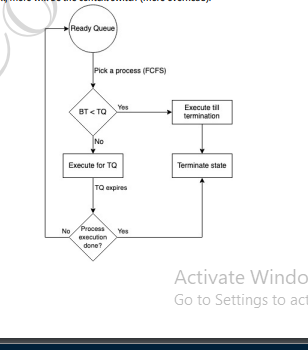
1. Shortest Job First(SJF):[Non – Pre-emptive)
   1. Process with least burst time will get CPU
   2. P1 p2 p3 ->?? BT = Calculation of BT of process by CPU using estimation – so this is drawback
   3. 
   4. Victim of convoy effect
2. Shortest Job First(SJF):[Pre-emptive)
   1. No convoy effect
   2. Less starvation
   3. Gives average WT less for a given set of processes as scheduling short job before a long one decreases the WT of short job more than it increases the WT of the long process
   4. 
3. Priority Scheduling
   1. Assigns priority to each job
   2. Non Preemptive:
      1. No time sharing
      2. Assingning Priority



* + 1. SJF is a special case of general priority scheduling with priority inversely proportional to BT
  1. Preemptive priority scheduling
     1. Current RUN state job will be preempted if next job has higher priority
     2. May cause indefinite waiting (Starvation) for lower priority jobs. (Possibility is they won’t get executed ever). (True for both preemptive and non-preemptive version)
        1. average wait time – 11.4
        2. Biggest drawback – indefinite waiting / extreme starvation
        3. Rumour - IBM- 7094 at mit [ON->1967->JOBSUBT] Lowest priotity remained until 1973
        4. Solution to indefinite waiting:
           1. Ageing Gradually increasing priority of lowest priority
           2. 15secs -> Lowest priority jobs (priority + 1)
        5. It has convey effect
  2. Round robin scheduling (RR):
     1. Most popular
     2. FCFS (Preemptive) version
     3. Criteria (AT + Time Quantum)
     4. Designed time sharing
     5. Easy to implement
     6. 
     7. No process is going to wait forever, hence very low starvation. [No convoy effect]
     8. If TQ is small, more will be the context switch (more overhead).
     9. 