Ans 1. FIFO CAPACITY SCHEDULER

FIFO: The FIFO Scheduler places applications in a queue and runs them in the order of submission (first in, first out). Requests for the first application in the queue are allocated first; once its requests have been satisfied, the next application in the queue is served, and so on. Large applications will use all the resources in a cluster, so each application has to wait its turn. On a shared cluster, it is better to use the Capacity Scheduler or the Fair Scheduler

CAPACITY: With the Capacity Scheduler, a separate dedicated queue allows the small job to start as soon as it is submitted. This is at the cost of overall cluster utilization since the queue capacity is reserved for jobs in that queue. Large job finishes late when compared with using the FIFO Scheduler.

Ans 2. FIFO FAIR SCHEDULER

FIFO: The FIFO Scheduler places applications in a queue and runs them in the order of submission (first in, first out). Requests for the first application in the queue are allocated first; once its requests have been satisfied, the next application in the queue is served, and so on. Large applications will use all the resources in a cluster, so each application has to wait its turn. On a shared cluster, it is better to use the Capacity Scheduler or the Fair Scheduler

FAIR: With the Fair Scheduler, there is no need to reserve a set amount of capacity, since it will dynamically balance resources between all running jobs. Just after the first (large) job starts, it is the only job running, so it gets all the resources in the cluster. The overall effect is both high cluster utilization and timely small job completion.

Ans 3. CAPACITY FAIR SHEDULER

CAPACITY: With the Capacity Scheduler, a separate dedicated queue allows the small job to start as soon as it is submitted. This is at the cost of overall cluster utilization since the queue capacity is reserved for jobs in that queue. Large job finishes late when compared with using the FIFO Scheduler.

FAIR: With the Fair Scheduler, there is no need to reserve a set amount of capacity, since it will dynamically balance resources between all running jobs. Just after the first (large) job starts, it is the only job running, so it gets all the resources in the cluster. The overall effect is both high cluster utilization and timely small job completion.

Ans 4.

LIMITATIONS OF 1.X : SOLUTION IN 2.X

1. Memory problem in hdfs due to so many jobs in the 1.x : HDFS FEDERATION which divides the memory and have the heart beat taken by each block of memory.

2. Back up problem : 2.x provides high availability which never rests in the time of crashing the active name node.

3.overloading problem in task tracker : 2.x provides yarn (yet another resource negotiator) under which 50 frame work.