Identical Machine Scheduling with Precedence Constraints

We are given a directed acyclic graph on $n$ vertices in which each vertex represents a unit time task. An edge $(u \rightarrow v)$ in the graph implies that task $v$ cannot be started until task $u$ has been completed. Each task can be executed on any of $m$ identical machines. The goal is to minimize the total time needed until all tasks are completed. An obvious lower bound on the value of $OPT$ is $n/m$ or the longest directed path in the graph. It is known that the greedy algorithm, that schedules a job if it is feasible, provides a 2 approximation.

Questions:

1. Is the problem NP-hard if $m$ is fixed?

2. What happens if preemption is allowed?