1 Explain why bounds are not significant when studying applications with sporadic tasks.

2 What does it mean that EDF is "optimal"? Can you show intuitively why that is the case?

3 Consider the real-time task scheduling problem with/without precedence constraints and with/without preemption. What are the easiest and the hardest cases? Why?

4 Consider a generic application composed of periodic and sporadic tasks, to be scheduled using static priorities. Can feasibility be determined using bounds?

5 Discuss optimal versus heuristic scheduling algorithms. When would you use either type of scheduling algorithm?

6 Explain the different types of blocking that can occur when PCP is used to regulate access to resources, and show why such blockings may occur.

7 Briefly present a method for scheduling aperiodic jobs with precedence constraints, assuming that jobs are non-preemptible and they may share resources. Can we say that this a difficult scheduling problem?

8 Compare the non-preemptive (NPP) and highest locker priority (HLP) protocols. Discuss cases when either protocol may be preferred to the other.

9 Discuss blocking time under priority inheritance: in what ways does it differ from that obtained under other resource access protocols?

10 Clock-driven scheduling is a very common method for scheduling periodic tasks. Briefly discuss the main ideas and what are its advantages/disadvantages with respect to priority-driven scheduling.

11 Discuss how precedence relations and the (im)possibility to preempt tasks affect the difficulty of scheduling problems.

12 Explain priority inversion, and why it is a problem in real-time systems.

13 The stack resource policy protocol (SRP) is a way of dealing with priority inversion. Discuss its advantages and disadvantages with respect to other resource access protocols seen.