(1) What do “necessary conditions” guarantee?

If a necessary condition is not met, it guarantees that you will not get what you are looking for.

(2) What are the four necessary conditions for a process deadlock to occur?

① Mutually-exclusive devices
② Non-preemptive devices
③ Hold & wait
④ Circular wait

(3) What are the two different types of “process deadlock” (those CS314 addresses)?

① Process deadlocks due to (improper uses of) semaphores
② Process deadlocks due to I/O resources
(4) In deadlock prevention, one of the solutions is not to allow any process to “hold & wait” (if a process that holds some non-preemptive resources needs additional resource(s), the process must release all the resources it currently holds and then it requests all the resources (both what it has released and what it additionally needs). Explain how this will prevent deadlocks.

To prevent process deadlocks by eliminating the condition of “hold & wait”, one way to do it is to enforce “all or nothing” requirement (if processes need more resources, they need to drop all the resources they hold and make requests to all resources at once – a process get them all (if all are available), or a process does not get any (if any is missing). This eliminates a chance of a process “holds & waits”.

(5) In the classroom, we discussed what we can do to make sure one of the four necessary conditions for a deadlock is not satisfied. Is it possible to have a technique that never causes “circular wait”? If yes, describe how.

One way to do it is by assigning a unique resource ID to each non-preemptive and mutually-exclusive resource and require all processes to make requests in the ascending (or the descending) order of the resource ID.